

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-317991

(43) 公開日 平成11年(1999)11月16日

(51) Int.Cl. ⁸		識別記号		F I	
H 0 4 Q	9/00	3 0 1		H 0 4 Q	9/00 3 0 1 E
G 0 6 F	3/00	6 5 1		G 0 6 F	3/00 6 5 1 A
H 0 4 N	5/44			H 0 4 N	5/44 A
	5/445				5/445 Z
	7/173	6 4 0			7/173 6 4 0 Z

審査請求 未請求 請求項の数21 F D (全 28 頁)

(21) 出願番号 特願平10-294728

(22) 出願日 平成10年(1998)10月2日

(31) 優先権主張番号 特願平9-271190

(32) 優先日 平9(1997)10月3日

(33) 優先権主張国 日本 (J P)

(31) 優先権主張番号 特願平9-343042

(32) 優先日 平9(1997)12月12日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 藤吉 靖浩

神奈川県横浜市磯子区新杉田町8番地 株式会社東芝マルチメディア技術研究所内

(72) 発明者 坂崎 芳久

神奈川県横浜市磯子区新杉田町8番地 株式会社東芝マルチメディア技術研究所内

(72) 発明者 安木 成次郎

神奈川県横浜市磯子区新杉田町8番地 株式会社東芝マルチメディア技術研究所内

(74) 代理人 弁理士 外川 英明

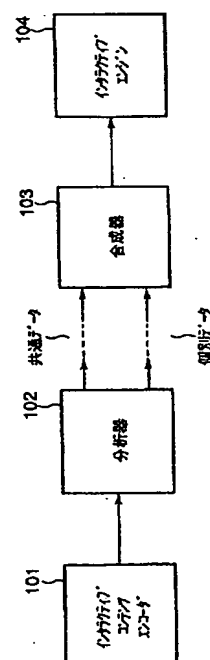
最終頁に続く

(54) 【発明の名称】 信号送信装置、信号受信装置、記録媒体

(57) 【要約】

【課題】それぞれ固有に開発された双方向システムに対し相互に互換性を確保することが可能な信号送信受信装置を得る。

【解決手段】実行するアプリケーションを、1つあるいは複数の方式に対応する、映像、音声、データとして送信する信号送信装置において、前記アプリケーションを構成する要素を、分析器102においてインターラクティブ機能を有する異なる方式で共通に用いる基本的な共通データと、おのおの異なる方式において個別に用いる個別データとに分けて送信し、受信側では、合成器103にて前記共通データと自身に必要な個別データとを合成し、自身のアプリケーションを完成させて実行する。



【特許請求の範囲】

【請求項1】 実行するアプリケーションを、1つあるいは複数の方式に対応する、映像、音声、データとして送信する信号送信装置において、

前記アプリケーションを構成する要素を、インタラクティブ機能を有する異なる方式で共通に用いる基本的な共通データと、おのおの異なる方式が個別に用いる個別データとに分解して送信することを特徴とする信号送信装置。

【請求項2】 実行するアプリケーションを、1つあるいは複数の方式に対応する、映像、音声、データとして送信する場合、前記アプリケーションを構成する要素を、インタラクティブ機能を有する異なる方式で共通に用いる基本的な共通データと、おのおの異なる方式が個別に用いる個別データとに分解して送信する信号送信装置からの信号を受信する受信装置であって、

前記信号送信装置の出力を受信し、前記共通データのみを用いてアプリケーションを実行するか、

または、前記信号送信装置の出力を受信し、前記共通データおよび、自身の方式に対応した前記個別データとを合成してアプリケーションを実行するか、

または、前記信号送信装置の出力を受信し、前記共通データおよび、自身の方式に対応した前記個別データ、更に、ひとつ以上の自身の方式と異なる方式に対応した前記個別データとを合成してアプリケーションを実行するか

のいずれかの手段を有する信号受信装置。

【請求項3】 前記共通データは、アプリケーションの構成を記述したオブジェクトと、所定の符号化方式に従ってエンコードされたコンテンツと、制御動作を記述したスクリプトのデータの種類を持つことを特徴とする請求項2記載の信号受信装置。

【請求項4】 前記共通データは、自身を識別させる第1の識別子のみを持つことを特徴とする請求項2記載の信号受信装置。

【請求項5】 前記個別データは、共通データの性質を基本的に有し、個別のテレビジョンシステムに独自で共通データに不足する詳細な情報である補足データと、共通データの性質を基本的に有しないスクリプト、コンテンツデータである独自データとからなり、

前記個別データ中の前記補足データのデータ単位は、自身の基本的な性質を持つ共通データを知るための第2の識別子を持ち、同独自データのデータ単位は、自身を識別するための第1の識別子を持つことを特徴とする請求項2記載の信号受信装置。

【請求項6】 第1の方式に適合するアプリケーションのための第1の個別データと第1の共通データを受信し、前記第1の個別データを変換することにより、第2の方式のアプリケーションに必要な第2の個別データを生成し、

この第2の個別データと前記第1の共通データとを用い

て前記第2の方式に適合するアプリケーションを実行する手段を有することを特徴とする請求項2記載の信号受信装置。

【請求項7】 表示装置で実行するアプリケーションを、1つあるいは複数の方式に対応する、映像、音声、データとして定義する場合、前記アプリケーションを構成する要素を、インタラクティブ機能を有する異なる方式で共通に用いる基本的な共通データと、おのおの異なる方式が個別に用いる個別データとに分解して定義し、この共通データと個別データとを夫々バケット化して多重して記録していることを特徴とする記録媒体。

【請求項8】 実行するアプリケーションを、1つあるいは複数の方式に対応する、映像・音声・データとして送信する場合、前記アプリケーションを構成する要素を、インタラクティブ機能を有する異なる方式で共通に用いる基本的な共通データと、前記異なる方式が各々個別に用いる個別データとに分解して送信する信号送信装置からの信号と、自身を操作するために用いるデータを含み、前記インタラクティブ機能を有する異なる方式に変換可能なフォーマットに従ったデータを送信する機能を有する1つ以上の装置からの信号を受信する受信手段を有し、

前記共通データ、前記個別データ、前記変換可能なフォーマットに従ったデータのいずれか1つあるいは、これら受信したデータのうち1つ以上のデータの合成データを用いてアプリケーションを実行する実行手段を有したことを特徴とする信号受信装置。

【請求項9】 前記変換可能なフォーマットに従ったデータを用いたアプリケーション実行の結果として、該データの送信元である装置自身を制御するためのデータを発生して送出する手段を有することを特徴とする請求項8記載の信号受信装置。

【請求項10】 前記送信元である装置は、ネットワーク機器であり、各々のネットワーク機器は自身が備えている機能を示す機能データを保持しており、前記実行手段は、前記機能データを参照し、前記各々のネットワーク機器を制御することを特徴とする請求項9記載の信号受信装置。

【請求項11】 前記各々のネットワーク機器が保持する前記機能データを前記実行手段が参照する場合、実行手段の種類に関わらず一定のインタフェースを通して参照されることを特徴とする請求項10記載の信号受信装置。

【請求項12】 前記実行手段は、ネットワーク機器が備えている機能のそれぞれに対応するグラフィカル・ユーザ・インタフェース・オブジェクトを保持しており、前記グラフィカル・ユーザ・インタフェース・オブジェクトを通してユーザが前記ネットワーク機器を操作できるように、前記ネットワーク機器が備えている機能についての操作画像を表示画面上に表示することを特徴とす

10

20

30

40

50

る請求項10記載の信号受信装置。

【請求項13】 前記操作画像を複数表示するとき、前記操作画像の配置を調整する手段を有することを特徴とする請求項12記載の信号受信装置。

【請求項14】 複数のオブジェクトの合成データに対応する複数の操作画像の表示位置、大きさの情報を元に、複数の操作画像の配置及び大きさを調整する手段を有することを特徴とする請求項12記載の信号受信装置。

【請求項15】 前記共通データは、前記変換可能なフォーマットに従って記述されていることを特徴とする請求項8記載の信号受信装置。

【請求項16】 動作条件と動作記述とから構成されるリンクオブジェクトへイベントが入力され、前記イベントが動作条件を満たしていた場合にのみ前記動作記述で示された動作を行う第2のエンジン方式のアプリケーションを、イベントの発生と手続き型プログラムの実行とが等価である第1のエンジン方式のアプリケーションへと変換する場合、

第2のエンジン方式のイベントを、当該イベントに付随するデータを引数とした第1のエンジン方式の手続き型プログラムの呼び出しに変換するイベント変換手段と、第2のエンジン方式のリンクオブジェクトを構成する動作条件を、前記呼び出しの対象となる手続き型プログラム中の引数比較部に変換する動作条件変換手段と、前記リンクオブジェクト中の動作記述を前記手続き型プログラム中の動作記述部へと変換する動作記述変換手段とを備えることを特徴とする請求項8記載の信号受信装置。

【請求項17】 前記引数比較部は、前記第2のエンジン方式のリンクオブジェクト中の動作条件を判定の条件とし、この条件と前記引数比較部を含む前記手続き型プログラムの引数として渡されたデータとを比較し条件が成立するか否かを判定することを特徴とする請求項16記載の信号受信装置。

【請求項18】 前記手続き型プログラムの動作記述部は、第2のエンジン方式のリンクオブジェクト中の動作記述と等価な動作が第1のエンジン上で行われるように記述されていることを特徴とする請求項16記載の信号受信装置。

【請求項19】 前記動作記述変換手段は、第2のエンジン方式のリンクオブジェクト中の動作記述の構成要素となりうる全て或いは一部の記述の各々について、それらと等価な手続き型プログラムをあらかじめ生成しておき、この生成しておいた手続き型プログラムを参照しながら動作記述部を生成することを特徴とする請求項16記載の信号受信装置。

【請求項20】 イベントの発生と手続き型プログラムの実行とが等価である第1のエンジン方式のアプリケーションを、動作条件と動作記述とから構成されるリンク

オブジェクトへイベントが入力され当該入力されたイベントが前記動作条件を満たしている場合のみ動作記述によって示された動作を行う第2のエンジン方式のアプリケーションへ変換する場合、第1のエンジン方式のイベントを、第2のエンジン方式のイベントとリンクオブジェクトとの組に変換する手段を有することを特徴とする請求項8記載の信号受信装置。

【請求項21】 第2のエンジン方式のイベントとリンクオブジェクトの組の生成にあたっては、生成されるリンクオブジェクトは、常に成立する動作条件と、手続き型プログラムを呼び出す動作記述とから構成されることを特徴とする請求項20記載の信号受信装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、デジタルテレビジョン放送、ネットワーク、CATVシステムなどに有用であって、映像、音声に更にデータを多重して伝送する信号伝送装置、受信装置、記録媒体に関するものである。

【0002】

【従来の技術】近年、半導体技術、通信技術の発達により、より高度な機能をもつテレビジョン(TV)受像機の開発が行われている。特に従来のTV受像機が単に映像、音声信号を視聴し楽しむという受動的なものであったのに対し、デジタル技術を利用し、映像、音声信号以外にデジタルデータを多重し、映像、音声以外の情報を付加したり、視聴しているTV番組に対してレスポンスをするような高機能のテレビ受像機が望まれ始めている。このような機能を実現するTVとして、International Application Published under the patent cooperation treaty (PCT) International Patent Classification: H04N 1/00, International Publication Number: 0 96/34466に記載されているCompact Graphical Interactive Broadcast Information System(文献1)がある。本文献にはTV番組に通常の映像信号だけでなく、新たなデータを多重し、双方向機能を実現する手法が述べられている。

【0003】これにより、インタラクティブコンテンツを映像、音声信号と多重して伝送することにより通常のTV放送ではできないインタラクティブ機能を実現している。

【0004】一方、双方向機能を実現する方法として、ISO/IEC 13522-5(文献2)に記載されているMHEG-5と呼ばれるものがある。MHEG-5では、文献1と同様にアプリケーションとしてシーンと呼ばれるページに相当するものが定義されており、様々なアプリケーションをオブジェクトにより構成することが可能である。また、オブジェクトはデータコンテンツと呼ばれる文献1のリソースに対応するものを同様に使用することでアプリケーションを構成することができる。ただし、

文献2による双方向システムでは詳細なスクリプトの定義はなく、イベントの発生に伴うオブジェクトの動作を記述することでアプリケーションのシナリオを作成している。

【0005】以上の様に、文献1に記載されている双方向システムと文献2に記載されている双方向システムとはアプリケーションを構成する際にオブジェクトと呼ばれる構成要素を使用することで非常によく似ている。しかし、実際のアプリケーションを作成する際に使用できるオブジェクトの種類が異なるためそれぞれのアプリケーションに互換がない。また、最終的に伝送する際には双方とも最終形態としてバイナリーレベルで伝送されるため、伝送フォーマットについても互換性が無い。

【0006】以上2つの文献に記載された双方向システムについて述べたが、さらには第3、第4の双方向システムも開発されることが予想される。このような状況下においてはそれぞれ個別のシステムとなり、別々のエンコーダが必要となる。

【0007】

【発明が解決しようとする課題】以上のように、近年多くの双方向のテレビジョン方式が提案されているが、それぞれのシステムでは、アプリケーションに互換性がなく、それぞれ固有のエンコーダ、デコーダが必要となる問題が生じている。また、これらのフォーマットを強制的に統一してしまうと、自由なオブジェクトの定義あるいはシステムの開発に制約となる。

【0008】また近年では、テレビジョン受像機をディスプレイとして使用し、これに家庭内の各種機器（VTR、DVDなど）が複数接続されて使用される。この場合、それぞれの機器がリモコンを有し、その操作が煩雑となっている。

【0009】そこでこの発明は、上記課題を解決する双方向機能を有する信号伝送受信装置を提供するものであり、従来の方式に対して大幅な変更、追加を行わずに、それぞれ固有に開発された双方向システムに対し相互に互換性を確保することが可能な装置を提供することを目的とする。

【0010】またこの発明は、ネットワークに接続された種々の外部機器についてのガイド（アイコン、制御パネル）を見ることが出来、このガイド表示を通じて遠隔操作および制御を行うことができる信号送信装置、信号受信装置、記録媒体を提供することを目的とする。

【0011】

【課題を解決するための手段】この発明は、複数の異なるシステムの各アプリケーションを作成する上で最低限必要となるデータであって各システムに共通な共通データと、異なるシステムの各アプリケーションを前記共通データと共に完成する上で必要となるデータであって、前記共通データを除く各システムの個別の個別データとを分離して伝送する。

【0012】各システムの受信装置では、上記分離されて伝送されて来た共通データと、少なくとも自身の個別データを用いて自身に必要なアプリケーションを完成、または他のシステムの個別データを変換して自身に必要なアプリケーションを完成させるように構成される。

【0013】このような信号伝送受信方法では、送信側では各システムに互換性のあるアプリケーションを伝送することができる。また受信装置では、各システムの持つコンテンツを相互に利用可能とし、送信受信で閉じていたサービスを拡大することが容易となる。当然、従来例で供給されるアプリケーションについても問題なく動作することができる。それぞれの独自に開発したアプリケーションに対し、他のシステムの持つ機能を利用することも可能となるため、現状のシステムの機能を向上させることもできる。また、強制的にオブジェクトの種類、システム等について強制的な制約を与えることがないため、自在に新たなシステムを開発することができ技術的な進歩を阻害することがない。

【0014】またこの発明の方式では、上記の基本的な共通データと、前記異なる方式が各々個別に用いる個別データとに分解して送信する信号送信装置からの信号の他にさらに、自身を操作するために用いるデータを含み、前記インタラクティブ機能を有する異なる方式に変換可能なフォーマットに従ったデータを送信する機能を有する1つ以上の装置からの信号を受信する受信手段を有し、前記共通データ、前記個別データ、前記変換可能なフォーマットに従ったデータのいずれか1つあるいは、これら受信したデータのうち1つ以上のデータの合成データを用いてアプリケーションを実行する実行手段を有するものである。これにより本体側からアプリケーションを通じて、本体に接続されている機器を操作あるいは集約的に1個所で制御することが可能となる。

【0015】

【発明の実施の形態】以下、この発明の実施の形態を図面を参照して説明する。

【0016】図1は、この発明に係る第1の実施の基本形態を示す。インタラクティブコンテンツエンコーダ101は、作成されたインタラクティブコンテンツをエンコードし、分析器102へ出力する。分析器102は、入力されたインタラクティブコンテンツを分解し、共通データと個別データとを出力する。

【0017】なお、個別データとしては、その呼び名はこれに限るものではなく拡張データ、付加データ、差分データ、補助データなど各種の呼び名が可能である。また共通データについても、主データ、親データ、基幹データなど各種の呼び名が可能である。

【0018】共通データおよび個別データはそれぞれ合成器103に入力され合成される。合成されたデータはインタラクティブエンジン104に入力され、エンジンによりアプリケーションが実行される。

【0019】共通データは、例えば下記の要素から構成される。

〔1〕共通オブジェクト

〔2〕共通データコンテンツ

個別データは、例えば下記の要素から構成される。

〔1〕継承差分成分

〔2〕スクリプト

これらそれぞれの詳細については後述する。

【0020】図2には、この発明による第1の実施例を実際の送受信システムに適用した構成を示す。オーディオビデオ（AV）エンコーダ201は、映像・音声信号をエンコードしデジタル信号列に変換する。一方、インタラクティブコンテンツエンコーダ202は、インタラクティブコンテンツをエンコードし、分析器203へ入力する。分析器203は、インタラクティブコンテンツを共通データおよび個別データに分析分離し、フォーマットエンコーダ204へ出力する。フォーマットエンコーダ204では、共通データと個別データを多重し、多重器205へ出力する。多重器205は、AVエンコーダの出力とフォーマットエンコーダ204の出力を多重し、その多重信号の伝送を行う。

【0021】共通データと個別データとを多重する方法としては各種の方法があるが、例えば各データをそれぞれパケット化し、各パケットに識別コードを付加し、各パケットを時間多重する方法が採用される。

【0022】受信側では、多重信号を受信し、分離器206により映像・音声とインタラクティブコンテンツ用のデータを分離する。

【0023】分離器206は映像・音声データをAVデコーダ210へ出力し、インタラクティブコンテンツ用のデータをフォーマットデコーダ207へ出力する。フォーマットデコーダ207は、入力データを共通データおよび個別データに分離し合成器208へそれぞれのデータを入力する。合成器208では、共通データと個別データとを合成し、インタラクティブエンジン209へデータを出力する。インタラクティブエンジン209では、入力されたデータよりアプリケーションを構築して、結果をAVデコーダ210へ出力する。AVデコーダ210では、再生した映像、音声とアプリケーションを実行再生し、映像とグラフィック画像を表示する。

【0024】図3は、図1に示した分析器102および合成器103の詳細を示す。

【0025】分析器102は、コンテンツ分離器302、共通データフィルタ303、個別データフィルタ304、データ変換器305、306、307、308、多重器309、310より構成される。

【0026】コンテンツ分離部302は、入力部301より入力されるインタラクティブコンテンツをデータコンテンツ、オブジェクト、スクリプトに分解する。データコンテンツは、データ変換器305により所定のフ

ーマットに変換され共通データコンテンツとして多重器309へ入力される。同様にスクリプトはデータ変換器308により所定のフォーマットに変換され多重器310へ入力される。

【0027】オブジェクトは、共通データオブジェクトと、継承差分成分とに分離される。即ち、共通データフィルタ303は共通データオブジェクトとして定義されている成分を抽出し、個別データフィルタ304は共通オブジェクトとして定義されていない成分を継承差分成分として抽出する。共通データフィルタ303の出力（共通データオブジェクト）はデータ変換器306により所定のデータに変換され共通データコンテンツと共に多重器309により多重され出力部311より共通データとして出力される。一方、個別データフィルタ304の出力（継承差分成分）はデータ変換器307により所定のデータフォーマットに変換されスクリプトと共に多重器310において多重され、出力部312より個別データとして出力される。

【0028】合成器103は、分離器323、324、データ変換器326、327、328、329、オブジェクト合成器330、コンテンツ合成器331を有する。入力部321より入力される共通データは分離器323により共通データコンテンツと共通データオブジェクトに分離される。共通データコンテンツはデータ変換器326によりデータコンテンツに変換され、共通データオブジェクトも同様にデータ変換器327により所定のフォーマットに変換される。

【0029】一方、入力部322より入力される個別データは分離器324によりスクリプトと継承差分成分に分離される。継承差分成分はデータ変換器328により所定のフォーマットに変換され、スクリプトはデータ変換器329により所定のフォーマットに変換される。ここで、データ変換器327とデータ変換器328の出力はオブジェクト合成器330により合成され最終オブジェクトとなる。またデータ変換器326より出力されるデータコンテンツと、オブジェクト合成器330より出力される最終オブジェクトと、データ変換器329より出力されるスクリプトとは、再度コンテンツ合成器331により最終的なインタラクティブコンテンツとなるよう合成され、出力部332より出力される。

【0030】以上の様に、本発明における信号送受信装置では、オブジェクトを共通データオブジェクトと継承差分成分とに分割して伝送している。これらの構成について図4を用いて詳しく説明する。通常、オブジェクトを基本とするインタラクティブシステムでは、アプリケーション作成のための基本的なオブジェクトをもっている。本発明ではオブジェクト指向の技術を導入し、基本的なオブジェクトからそれぞれのシステムに固有なオブジェクトを生成する。オブジェクト指向の考え方では、オブジェクトは親となるオブジェクトの機能を継承し、

10

20

30

40

50

子となるオブジェクトを生成する。いずれの双方向システムにおけるオブジェクトにおいても、基本的オブジェクトの機能とその拡張として分解が可能である。文献1および文献2におけるオブジェクトについても例外ではない。

【0031】図4に示すように、インタラクティブアプリケーションのための最低限必要機能を持つオブジェクトをオブジェクトコアクラス401とすると、いずれのアプリケーションにおいても、アプリケーション全体を表す大局的な性質を持つグローバルクラス402、グラフィックとして表示されるビジュアルクラス403、音等の見えないが再生されるインビジュアルクラス404、内部的な動作をつかさどるインターナルクラス405等の抽象的なクラスに分類できる。これら抽象的なクラスから継承してできる具象化クラスの例を406から414に示す。即ちアプリケーション、フォーム、ビットマップ、テキスト、図形、エントリーフィールド、サウンド、イベント、バイトコードである。これらはいずれのインタラクティブアプリケーションを作成するための最低限必要なオブジェクトの例である。この具象化クラスまでを共通データオブジェクトとし、異なるシステム特有のオブジェクトはこの共通データオブジェクトから継承することで生成することができる。継承差成分は、共通データオブジェクトから各システムのオブジェクトを継承した時の差を示す成分である。

【0032】この実施の形態では、上記の共通データオブジェクトとして下記の9つを示している。

【1】アプリケーション：アプリケーション全体に必要な情報を持つ。

【2】フォーム：1枚のページを構成するための情報をもつ。

【3】ビットマップ：画像を表示する。

【4】テキスト：テキストを表示する。

【5】図形：矩形やラインを表示する。

【6】エントリーフィールド：文字を入力する

【7】サウンド：音を再生する。

【8】イベント：オブジェクトの動きを制御する。

【9】バイトコード：スクリプトおよびバイトコードに関する情報をもつ。

【0033】図5には、具体的なオブジェクトの構造例を示す。

【0034】オブジェクトコア401として、オブジェクトID501およびタイプ502が生成される。ビジュアルクラスとして、座標情報503およびデータコンテンツへのコンテンツポインタ（あるいはID）504が生成され、ビットマップクラス408によりスクリプトへのスクリプトポインタ505が生成される。共通オブジェクトは以上の構成をとる。

【0035】以降は、システム特有の機能としての継承差成分が配置される。例えばビットマップ1では、色

や線に関する情報506が設定される。継承差成分は、この色や線に関する情報506である。これら全体はオブジェクトとして定義されている。

【0036】コンテンツポインタ504で指示されている位置には、データ列がデータコンテンツであることを表すID511を先頭にエンコード情報512、データコンテンツ本体513が配列されている。またスクリプトポインタ505で指示されている位置には、例えばバイトコードであることを表すID531を先頭にエンコード情報532、バイトコード本体533が配列されている。

【0037】データコンテンツに関しては、代表的なものとして、画像があげられるが、これらは通常圧縮されて伝送される。したがって、送信側で圧縮処理が行われエンジン側で伸張が行われている。オブジェクトの場合と異なり、画像圧縮は標準化がなされており、JPEG、GIF等のフォーマットで伝送される。したがって、データコンテンツに関しては、共通化が比較的容易にできるため、共通データコンテンツとして伝送が可能である。独自の圧縮方式等をとる場合は、共通データコンテンツと別に、個別データとして伝送することも可能である。この場合は、個別データコンテンツとしてスクリプトと多重して伝送される。

【0038】図6は、この発明による更に他の実施の形態である。

【0039】入力部601より入力されたインタラクティブコンテンツは分析器602において共通データ、個別データとに分解される。実際の伝送は、第2図に示すように、そのほかのデジタル信号と多重されて伝送される。

【0040】共通データは、合成器603を介して第1のエンジン604に供給される。また、共通データは、合成器605を介して第2のエンジン606にも供給され、合成器607を介して第3のエンジン608、合成器609を介して第4のエンジン610に供給される。

【0041】個別データは、第1の個別データとして合成器605に入力され、またデータ変換器611で変換され第2の個別データとなり合成器607に入力される。更にまた第1の個別データは合成器613に入力されると共に、データ変換器612で変換されて第2の個別データとなり、合成器613に入力されている。この結果、合成器613からは第1の個別データと第2の個別データを合成した第3の個別データが得られ、この第3の個別データは合成器609に入力される。

【0042】この実施例では、第1のエンジン604は、共通データだけを受信しデコードする。第2のエンジン606は、共通データと第1の個別データとを受信しデコードする。また第3のエンジン608は、共通データと第1の個別データから変換された第2の個別データとを受信しデコードする。更に第4のエンジン610

は、第1の個別データを第2の個別データに変換し、この第2の個別データと第1の個別データを合わせることで得られた第3の個別データと、共通データを受信してデコードを行なう。

【0043】第1のエンジン604は、共通データだけを受信するため、機能は限られているが異なるシステムでエンコードされたインタラクティブコンテンツを受信しアプリケーションを実行することが可能である。第2のエンジン606は、本来のアプリケーションを再生することができる。第3のエンジン608は、本来異なるシステムのエンジンであるが、第1の個別データを第2の個別データへ変換して用いることにより異なるシステムのアプリケーションを実行することが可能となる。データ変換器611は、オブジェクトの継承差成分のみを変更することで、実現が可能であるため、オブジェクト自体（オブジェクトの全体）を変更する変換器に比べて小規模に実現することができる。第4のエンジン610の系統についても同様ことが言える。

【0044】図7には更に別の実施の形態を示している。

【0045】各システムの装置が独立して構成される場合、システム1ではエンコーダ702と第2のエンジン718が対応関係にあり、システム2ではエンコーダ703と第3のエンジン719が対応関係にあり、システム3ではエンコーダ704と第4のエンジン720が対応関係にある。この実施の形態では、異なるシステム間で互換性を持たせた構成を得る。

【0046】オーサリングツール701により出力されたインタラクティブコンテンツは、エンコーダ702、703、704によりエンコードされる。それぞれのエンコードされたデータは、分析器705、706、707に入力され共通データと、第1、第2、第3の個別データに変換される。共通データはすべて同一であるため、分析器706、707からは第2、第3の個別データのみを導出する。多重器708は、共通データと第1乃至第3の個別データを多重し出力部709より出力する。

【0047】受信側では、入力部711より入力されたデータを分離器712により共通データと、第1の個別データと、第2の個別データと、第3の個別データに分離する。

【0048】共通データは、合成器713、714、715、716、723に共通に入力される。ここで、合成器713は、共通データのみによりインタラクティブコンテンツを生成し、アプリケーションを実行する第1のエンジン717に供給する。合成器714は、共通データと第1の個別データによりインタラクティブコンテンツを生成し、アプリケーションを実行する第2のエンジン718に供給する。合成器715は、共通データと第2の個別データによりインタラクティブコンテンツを

生成し、アプリケーションを実行する第3のエンジン719に供給する。合成器716は、共通データと第3の個別データによりインタラクティブコンテンツを生成し、アプリケーションを実行する第4のエンジン720に供給する。

【0049】このように、異なるシステムに対応したエンジンにおいても、アプリケーションを実行することが可能となる。

【0050】ここで、さらに、合成器722により第2、第3の個別データを用いて第4の個別データを生成し、合成器723により第4の個別データと共通データを合成することで、新たな第5のエンジン724では、エンジン719およびエンジン720のアプリケーションの機能を兼ね備えたエンジンを構築することが可能となる。

【0051】本実施の形態では、個別データが3つに増えているため、より大きな伝送容量が必要に見えるが、通常アプリケーションでは、画像データ等の共通データコンテンツがインタラクティブコンテンツのほとんどの部分を占めているため、個別データの増加は無視可能できるほど小さい。

【0052】図8に、上記した実施の形態で示すシステムのアプリケーション例を示す。第1のエンジン717のアプリケーションが実行された場合の画像状態を図8(a)に示し、第3のエンジン719のアプリケーションが実行された場合の画像状態を図8(b)に示し、第5のエンジン724のアプリケーションが実行された場合の画像状態を図8(c)に示している。

【0053】図9には、共通データ、第2のエンジン用の第2の個別データ、第3のエンジン用の第3の個別データの例を示している。第1のエンジン717は、共通データのみによりアプリケーションを実行する。共通データでは、基本的なパラメータのみが定義されている。第2の個別データにおける継承差成分には、ボタンおよびビットマップの形状に楕円を指示する属性が含まれるため、第2のエンジン719により実行されるアプリケーションでは、図8(b)に示すように、楕円の形状となる。さらに、第3の個別データでは、ビットマップオブジェクトに半透明の属性が含まれているため、第5のエンジン724では、第2、第3の個別データの両方の継承差成分が利用可能となるため、楕円がかつ半透明のビットマップを生成することができる。

【0054】第10図には、共通データと第2と第3の個別データを多重したデータ構造を示す。ヘッダ情報に続けて共通データ部の先頭にインジケータとして“エンジン1”が示されている。さらに、オブジェクト、データコンテンツが挿入されている。続けて、個別データ部分は、第2の個別データを示すインジケータとして“エンジン2”が示され、個別データとしての継承差成分が挿入される。さらには、第3の個別データを示すイン

ジータ“エンジン3”がつづき継承差分成分が挿入されている。

【0055】図11を参照しながら、上述のアプリケーション構成要素の伝送方法について説明する。

【0056】図11は、この発明の一実施の形態におけるデータ伝送単位層構造説明の補助図である。図11

(a)のアプリケーション要素(application element)は、1つのアプリケーションの全ての構成要素を含む伝送単位である。その構造は、アプリケーション全体に関係する情報を持つアプリケーション要素ヘッダ(application element header) 1101と、各エンジン用のデータを含むコンテナ(container) 1102, 1103, 1104からなる。

【0057】第1のエンジン用のコンテナ1102は、各エンジンが共通に使うデータ、また、第2のエンジン用のコンテナ1103は、コンテナ1102に含まれない第2のエンジン専用のデータ、第3のエンジン用のコンテナ1104は、コンテナ1102に含まれない第3のエンジン専用のデータを伝送するものとなっている。各コンテナが、どのエンジン用のデータであるかは、コンテナヘッダ(container header) 1105内のブロードキャストアプリケーションタイプ(Broadcast Application type: BAPタイプ) 1110で識別する。ひとつのアプリケーション要素には任意の数のコンテナを持つことが出来る。各コンテナは、図11(b)で示すような共通の構造を持つ。その構造は、そのコンテナ全体に関係する情報を持つコンテナヘッダ1105、コンテナ定義付け部(container definerecord) 1106と、アプリケーション本体のデータを伝送するオブジェクト(objects)、コンテンツ(contents)、スクリプト(script)の3種のレコードから構成される。

【0058】レコードは、上記3種のデータに分けて、その構成要素を書き並べた形態となっている。例えば、オブジェクトレコード(object record)は図11

(c)に示すように、レコードヘッダ(record header) 1111とアプリケーションを構成する全オブジェクトである、オブジェクト1, 2, ..., n, ..., Nから構成される。コンテンツレコード(content record)およびスクリプトレコード(script record)も同様に設定されるが、図11ではオブジェクトレコードを代表して記載している。図9に示した第1のエンジン用のデータを送る場合であれば、オブジェクトは、シーン(Scene)、イベント(Event)、ビットマップ(1)(Bitmap(1))、ボタン(2)(Button(2))、ボタン(3)(Button(3))が構成要素となる。同様にコンテンツレコードでは、ビットマップ(1)(Bitmap(1))、テキスト(2)(Text(2))、テキスト(3)(Text(3))が構成要素となり、スクリプトレコードでは、スクリプトA(2)(scriptA(2))、スクリプトA(3)(script A(3))が構成要素となる。

【0059】レコードがどのデータ種別のものかは、レコードヘッダ1111中のレコードタイプ1116で識別する。もちろん、そのエンジン用に何れかの種類のデータを持たない場合、そのレコードは、伝送されない。例えば、図9で説明した第3のエンジン用のデータでは、ビットマップオブジェクト(bitmap object)のみ存在するので、第3のエンジン用のコンテナでは、オブジェクトレコードのみが存在する。オブジェクト、コンテンツ、スクリプトは各々識別情報(ID) 1117, 1120, 1123とその種別を示す種別情報(type) 1118, 1121, 1124、実際のデータとしてのボディ(body) 1119, 1122, 1125を持つ。ここで種別とは、図9の例をとるとすれば、オブジェクトレコード内ではシーン(Scene)、イベント(Event)、ビットマップ(Bitmap)、ボタン(Button)であり、コンテンツレコード内では、ビットマップ(Bitmap)、テキスト(Text)であり、スクリプトレコード内では、スクリプトA(script A)である。

【0060】IDは、コンテナとレコードの種別によりその意味が異なるので、その内容について図4、図12を参照しながら以下に説明する。

【0061】IDは、自身の識別番号を意味するものと、自身の継承元を示す番号である場合とに分ける事が出来る。図12にコンテナとレコードの種別毎のIDの意味を示す。上述して来たとおり、この発明では、任意のエンジンは、各エンジン共通の共通データと、この共通データに無い自身に独自の個別データとを組み合わせる。各々のデータは別のコンテナで伝送されるため、受信後にそれらを足しあわせることになるのでその関係が明示される必要があり、このためにIDが用いられる。

【0062】図4に示したように、個別エンジン用の個別データであるビットマップオブジェクト415, 416は、共通データであるビットマップオブジェクト408と足し合わされる。この時、ビットマップオブジェクト415, 416の持つIDは、自身の継承元である共通データのビットマップオブジェクト408のIDと同じ値を持ち、その継承関係を示している。

【0063】このため、図12に示す通り、第1のエンジン用のコンテナ中のオブジェクトレコード内のオブジェクトは、自身を識別する番号としてIDを持ち、他のエンジン用の差分データを送るコンテナ中のオブジェクトレコード内のオブジェクトは、自身の継承元(所属)となる前記第1のエンジン用のオブジェクトのIDと同じ値のID値を持つ。また、これら継承関係を示すためのID値は全く同じ値を持つ他、例えばID値の一部(例えば、上位数ビット)で継承関係を示しても良い。

【0064】一方、コンテンツやスクリプトは足しあわせて使われることが無いので、全ての場合に自身を識別する番号としてそれぞれ異なるID値を持ち、オブジェ

クト内から、そのID値で参照される。

【0065】次に、図13～図18を参照しながら、本発明のデータ伝送の一実施の形態について説明する。

【0066】図13は、“ICAP Specification, Version 1.0, Document Revision 1.2”, Apr. 1997（文献3）に従って構成された第2のエンジン用のボタンオブジェクトの構成であり、図14は、文献2に従って構成された第3のエンジン用のボタンオブジェクトの構成である。また、図15はこれらのビットマップオブジェクトを本発明によるデータ伝送方式により構成したものである。

【0067】図13の“object Type”と図14の“object-identifier”は共にオブジェクトの識別子を表すので、図15の共通要素“Object ID”としている。同様にオブジェクトの表示位置を表す図13の“position”, “extent”と、図14の“x-length”, “y-length”, “x-position”, “y-position”は、図15の共通要素“UpperLeft Position”と、“LowerRight Position”とし、共にボタンデータへのポインタを示す図13の“resource”と図14の“content-data”は図15の“Content Reference”としている。これら以外の部分はそれぞれ図15の個別データである差分要素2と差分要素3として構成される。ボタンを丸く表示することを示すフラグである“hints:HINT_ROUND_KEYCAP”は差分要素2の中に挿入して構成される。

【0068】図16は、第2のエンジン用のビットマップオブジェクトの構成であり、図17は、第3のエンジン用のビットマップオブジェクトの構成である。また、図18は、これらのビットマップオブジェクトを本発明によるデータ伝送方式により構成したものである。

【0069】図16の“object Type”と図17の“object-identifier”は共にオブジェクトの識別子を表すので、図18の共通要素“Object ID”としている。同様にオブジェクトの表示位置を表す図16の“position”, “extent”と図17の“x-length”, “y-length”, “x-position”, “y-position”は図18の共通要素“UpperLeft Position”と“LowerRight Position”としている。また、共にボタンデータへのポインタを示す図16の“resource”と図17の“content-data”は、図18の“Content Reference”としている。

【0070】これら以外の部分はそれぞれ図18の差分要素2と差分要素3として構成される。ビットマップを半透明で表示することを指示するフラグである“transparency”は差分要素3として構成される。

【0071】さて、上記実施例における共通データの構成は、異なるシステムのすべてに共通なデータ概念（例えば「テキスト」や「ビットマップ」など）が存在することが前提となっているが、そのような場合だけでなく、異なるシステム間においては、あるシステムには存在するデータ概念が、他の一つ以上のシステムには存在

しないという場合も考えられる。このような場合には共通データとして構成できるデータが存在しないためにそのままでは上記の手法は適用できないが、ボディーのない空の共通データを定義することによって、先に述べたこの発明のデータ構成法を維持しながら用いることが可能である。

【0072】空の共通データは、ボディーがなくIDのみを持つ共通データであり、これを用いるとシステム固有のデータはその全体が個別データとして構成される。

10 個別データには、空の共通データのオブジェクトID（object ID）が割り当てられ、空の共通データが継承元であることを示す。

【0073】例として、第2のエンジンだけに存在し、その他のエンジンには存在しない第2のエンジン固有のオブジェクトを上記手法によって構成する場合について図19と図20を用いて説明する。

【0074】ここではオブジェクトについて示しているが、コンテンツおよびスクリプトの場合も同様に構成できる。図19においては、アプリケーション要素3001（図19（d））に含まれる第1のエンジン（共通データ）用のコンテナ3002（図19（e））中のオブジェクトレコード3003にはオブジェクトが並んでおり、このうちの一つオブジェクト3004はオブジェクトタイプ（object type）が“NULL（空）”であり、オブジェクトIDのID値＝2である空のオブジェクトが置かれている（図19（f）、（g））。このオブジェクト3004にはオブジェクトボディー（object body）がない（データの中身が空である）。

【0075】一方、第2のエンジン固有のオブジェクトの実体は、その全体が、第2のエンジン用のコンテナ3005（図19（c））中のオブジェクトレコード3006にはオブジェクトが並んでいる。この並んでいるうちの一つのオブジェクト3007はオブジェクトタイプが“第2のエンジン固有”であり、オブジェクトIDのID値＝2となっている（図19（a）、図19（b））。このように第2のエンジン用のオブジェクトIDのID値は第1のエンジンのものと同じくID値＝2となっており、空オブジェクト3004を継承することを示している。

40 【0076】図19の例では、NULLオブジェクトを各種種類のレコード（オブジェクトレコード、コンテンツレコード、スクリプトレコード）の中に置いたが、図20に示すように、第1のエンジン用のコンテナ3011中（図20（d））のオブジェクトレコード、コンテンツレコード、スクリプトレコード以外に、空の共通オブジェクトだけを集めた第4の種類の空のレコード3012を定義し、ここに空の共通データを置く構成としてもよい（図20（e））。図20の例では、継承先のオブジェクト（図20（a）、（b）、（c））は、継承元のオブジェクト（図20（g））を継承するよう、ID値

= 2 になっている。

【0077】上記のように空のオブジェクトと個別オブジェクトから構成されたデータを第2のエンジンが受け取ると、第2のエンジンは、共通データ（第1のエンジン用のコンテナ）の中の空オブジェクトからオブジェクトIDのID値を取り出し、自身の個別データ（第1のエンジン用のコンテナ）から上記取り出したオブジェクトIDのID値を持つ個別オブジェクトを取り出し、合成する。この合成した結果、空オブジェクトのIDを持ち、実際には個別オブジェクトのみのボディーを持った第2のエンジン用のオブジェクトが構成される。第2のエンジン以外のエンジン、例えば第3のエンジンが上記データを受け取ったとしても、共通データの中から空オブジェクトを取り出すものの、自身の個別データ（第3のエンジン用のコンテナ）の中にはこの空オブジェクトが持つIDのID値に対応する個別オブジェクトがないため、この空オブジェクトは何も処理されない。

【0078】次に図21を参照しながら空の共通データを用いる場合に使用する空の共通データの数を減らす方法を説明する。空の共通データは、オブジェクト、コンテンツ、スクリプトといったデータの種類を持っておらず（図21（a））、合成された結果のデータの種類の合成される個別データ（図21（b））によって決まる。

【0079】もしこの空の共通データと関連付けられる個別データが、あるシステムの1つだけに固有であり、また、別の空の共通オブジェクトが、別のシステムの1つだけに固有な個別データに関連付けられている場合は、空の共通オブジェクトを2つ使用しなければならない。しかしこの発明では、これらの空の共通データを共有することができる。すなわち、図21に示すように、第2のエンジンの中にある個別データ2と、第3のエンジンの中にある個別データ3の両者が、共に同じID値（例えば5）を持っている。第2のエンジンは空の共通データと個別データ2からID値が5のデータを生成し、第3のエンジンは空の共通データと個別データ3からID値が5のデータを生成する。ここで個別データ2は第2のエンジンの中に固有であり、個別データ3は第3のエンジンの中に固有であるため、衝突は生じない。このような手法によって共通データレコードに埋め込む空の共通オブジェクトの数を減らすことができ、すなわち共通データのデータIDの発行数を減らすことが可能になる。

【0080】以上、システムに固有のデータ概念を空の共通データを用いて構成する方法を示したが、空の共通データのIDのID値が必ず個別データのIDのID値として記されることから、空の共通データを省略する構成としてもよい。

【0081】空の共通データを省略すると、省略しない場合には空の共通データへの参照という意味を持ってい

る個別データのIDが、その個別データそのものを識別するIDとしての意味に変わる。したがって個別データ中の他の通常のデータ単位すなわち継承元となる共通データを参照している補足データとセマンティックスが変化するが、シンタックスそのものは全く同様である。

【0082】上記した実施の形態では、共通データと個別データとを伝送装置により伝送し、受信側では共通データと、自身に必要な個別データとを用いて自身のアプリケーションを構築する送受信方式として説明した。しかしこの発明は、このような送受信方式に限定されるものではない。例えば、共通データと個別データが記録された光ディスク、あるいは磁気ディスクなどの記録媒体を構成してもよい。ユーザは、この記録媒体を再生装置で再生して、共通データと個別データとを用いて再生装置側で自身に必要なアプリケーションを構築するようにしてもよい。また別の実施の形態として、共通データと個別データが伝送されて来た場合、一旦このデータを全て記録媒体に格納してもよい。そしてこの記録媒体に格納されている共通データと個別データとを用いて自身に必要なアプリケーションを構築するようにしてもよい。このようにすると、実際にアプリケーションを利用する時間帯と、共通データおよび個別データが伝送されてくる時間帯とが異なってもよい。

【0083】上記した実施の形態は、放送センター側から共通データと個別データが伝送されてくることを主体にして説明したが、これに限るものではない。たとえば、家庭内に設置されている他の機器からも受信側には、データが伝送されても本発明は適用できるものである。たとえば家庭内の他の機器としては、ビデオテープレコーダ、デジタルビデオディスク再生装置、ディスク再生装置、パーソナルコンピュータ、エアーコンディショナー、冷蔵庫、他のテレビジョン受信機等があり、これら機器からのデータもネットワークを通じて伝送される。

【0084】図22を参照しながら、実施の形態を説明する。ただしネットワークのトポロジに関してはこの限りではない。

【0085】図22において、送信装置800では、オーサリングツール（Authering Tool）801で制作されたアプリケーションは、分離器802でアプリケーションの実行方式に依らず共通に用いる共通データと、実行方式により使用の可否が分かれる個別データに分離される。この後、共通データは、変換器803で前記実行方式に依らず所定のフォーマットに変換され、個別データと共に多重器804に入力されて1つのデータとされる。このデータは、図示しない画像や音声のデータと更に多重され、これも図示しない所定の送信の為の処理を施された後、送信端子805から送信される。送信されたデータは、電波による放送などの伝達手段を介して、受信装置900で受信される。受信装置900は、グラ

フィカル・ユーザ・インターフェイスを構築するのに必要な表示画面908を常に所有しなければならない。

【0086】受信装置900の入力端子901から取り込まれたデータは、図示しない受信後の所定の処理を施された後、これも図示しないが、画像、音声のデータと分離された後、分離器902に導かれ、共通データと個別データに分離される。分離された共通データは、変換器903で実行方式に準じたフォーマットに変換された後、個別データと共に合成器904に入力される。

【0087】一方、ネットワーク装置951（例えばVTR）、952（例えばDVD）および953（例えば第2VTR）は、自身の制御パネルを模したグラフィックスや動作を実現するスクリプト等のオブジェクトを、例えばIEEE1394の様なネットワーク960を介し、受信装置900の入出力端子920に供給する。

【0088】受信装置900では入出力端子920からの入力をインターフェース（I/F）905へ導き、エンジン906に直接与えることができると判断されるデータ（共通データに相当）は直接エンジン906へ導き、そうでないデータ（個別データに相当）は、変換器907で実行方式に則したフォーマットへ変換した後、合成器904で前記2つの入力と合わせて合成し、そのデータをエンジン906に与える。ここで、実行方式と呼んでいるのは、すなわちエンジンの構成を決めている方式（エンジンが正しく処理可能な方式）のことである。

【0089】エンジン906では、合成器904およびインターフェース905の出力であるアプリケーションデータを実行し、送信装置800からのアプリケーションをネットワーク装置951、952、…からのオブジェクトを交えて表示器の表示画面908に表示する。

【0090】ここで図示しないリモートコントローラなどの外部入力装置により、該オブジェクトが選択されると、前記スクリプトがバーチャルマシン（VM）909で実行され、この結果が、エンジン906、I/F905、入出力端子920を介してネットワーク上の該スクリプトを受信すべきネットワーク装置へ伝達され、該装置が所定の動作を行う。

【0091】図23を参照しながら、さらに上記アプリケーションの具体的な動作例を説明する。

【0092】図23は、例えば受信装置の電源をオンしたときにアプリケーションが表示画面908に表示された例を示している。この実施例では、放送番組マテリアルと、ネットワーク装置が再生可能なマテリアルからユーザが希望のマテリアルを選択するアプリケーションを例に説明する。アプリケーションを表示画面908上に表示したときの基本的な画面レイアウトなどは、送信装置800から受信したデータによって決められている。しかしこれによらずユーザが任意に設定できるようになっていても良い。表示画面908の左側にはテレビ放送

選択窓2100、右側にはネットワーク機器選択窓2200が用意される。

【0093】テレビ放送選択窓2100の中には、放送を受信している各チャンネル（番組）の画像内容が、画面2101、2102、2103、2104のように表示される。また、ボタン2105、2106も表示される。このボタン2105、2106はスクロールボタンであり、他のチャンネルの画面を窓内に表示させる時に選択されるものである。各画面2101～2104も画像表示機能と共にボタン機能を持ち、ユーザが例えば画面2102を選択すれば、図に示すように画面の枠が強調される。また、図示しないが、表示画面908全体に画面2102（チャンネル3）の画像を表示するなどの機能を持つ。このような操作は、例えばリモートコントロール装置のシフトキーを操作すると、画面枠の強調部が次々と他の画面枠に移動し、希望のチャンネルの画面の枠が濃くなったときに決定キーを操作する。すると選択した画面のチャンネルが表示画面908の画面一杯に表示される。

【0094】一方、ネットワーク機器選択窓2200の中にも画面2201、画面2202が表示されている。この画面はネットワーク装置951、952が接続され電源オン状態で、かつ再生状態にあり、再生映像信号がネットワークを介して送られているとき、この再生されている映像を表示する。

【0095】ネットワーク装置951、952が接続され、かつ電源オン状態で待機状態にあるときは、ネットワーク装置をエンジン906が認識し、表示画面には例えば点線で画面枠、ボタン等による操作するための画像を表示する。

【0096】また、画面2201が画像表示状態にないとき、すなわち対応するネットワーク装置951が再生状態にない時は、対応するボタン2211を選択し、次にその内部の再生マークを選択し決定することで、再生を命令し画面2201に画像を表示させることができる。ここで、ネットワーク機器選択窓2200上の画面2201、2202、また、ボタン2211、2212の表示位置は、前述の通り、送信装置800から受信したデータまたは後述する表示位置調整方法等によって決められているが、画面に表示される画像データ、ボタン2211、2212の機能、形状データ、などは、対応するネットワーク装置から、ネットワークを介して得たオブジェクトを使うことも可能である。もちろんこの場合は、ネットワーク装置にあらかじめ自分の操作用のオブジェクト（操作するための画像を作成するためのビットマップや、操作するための制御コマンド）をデータとして保持させておき、電源がオンされ、かつネットワークに接続されたときは供給できるように設計される。このとき、受信装置900からのコールは有っても無くてもどちらでも良いように設計されている。

【0097】また、電源がオンされていなくても、ネットワークにネットワーク装置951が接続されたときは、受信装置900はネットワーク装置951からネットワークを介してオブジェクトを持ってきて使用することが出来るようになっていても良い。

【0098】図24に各ネットワーク装置と、受信装置900との関係を示している。

【0099】オブジェクトはその所有されている装置の一般的な、また固有の制御情報（機器を制御するための情報）とかつそれらを扱うためのGUI情報から構成される。受信装置900が立ち上がっている状態で、ネットワーク装置951が立ち上がるとネットワーク装置951はあらかじめその内部のオブジェクト記憶部962に所有しているオブジェクトを外部インタフェース961を通して、例えばIEEE1394等のネットワークを経由し、受信装置900のエンジン906に送出する。同様にネットワーク装置952その内部のオブジェクト記憶部972にオブジェクトを所有し、外部インタフェース971、ネットワークを通しエンジン906に送出する。

【0100】どちらの場合もエンジン906では、認識した各装置のオブジェクトをデコードしてアプリケーションを実行し、必要によりGUI情報を基に作成したアイコンや操作するための画像を表示画面908に表示する。また、逆に既にネットワーク装置951が立ち上がっており、後で装置952が立ち上がった場合も、ネットワーク装置951、952からオブジェクトを受けとり、先ほどの場合と同様に、オブジェクトをデコードしアプリケーションを実行する。

【0101】このエンジン906は先に述べた通り、伝送されてきた共通データと個別データとを全て図示しない記録媒体に記録する。そしてこの記録媒体に記録されているデータを用いて自身に必要なアプリケーションを構築しても良い。このようにすると、実際にアプリケーションを利用する時間帯と、共通データおよび個別データが伝送されてくる時間帯とが異なっても良いし、各装置が立ち上がる度に毎回同じオブジェクトを送ってもらわずに済む。

【0102】また、電波放送などで送信されたデータは、分離器902で共通データと個別データとに分離され、先の実施形態で説明したように、変換器903、907および合成器904から構成される変換合成部に渡されオブジェクトの変換合成が行われた後、このデータ・オブジェクトがエンジン906に渡される。このデータ・オブジェクトはエンジン906内で前述のオブジェクトと互換性のある形式に変換され、前述のオブジェクトと同様に扱われる。

【0103】図25には、エンジン906の内部において、オブジェクトのGUI配置を調整する機能を取り出して示している。ネットワーク装置から送られたオブジ

ジェクトは、オブジェクト検出器4010に渡され、オブジェクト検出器4010は入力を受けて、該当するネットワーク装置が立ち上がったことを画面に表示するため、該当するネットワーク装置のアイコンや操作するための画像であるグラフィックスを作成し、オブジェクト配置情報検出器4013およびオブジェクト配置調整器4014に送出する。

【0104】また、オブジェクトは合成器4011にも送られ、電波放送等で送信されたオブジェクトと合成される。この例では電波放送によりオブジェクトが送信され合成されるが、電波放送により送信されるオブジェクトは必ずしも必要ではなく、ネットワーク装置からのみオブジェクトが送られてくるだけであっても特に問題は無い。合成器4011は、合成したオブジェクトをエンジンコア4012に送出し、ここで、オブジェクトを適切に処理し、GUIをオブジェクト配置情報検出器4013とオブジェクト配置調整器4014に送出する。

【0105】オブジェクト配置情報検出器4013は、現在扱っているGUIの位置、大きさ、等を検出し、そのデータをオブジェクト配置調整器4014に送出する。オブジェクト配置調整器4014はオブジェクト配置情報検出器4013からのデータとあらかじめ設定した、またはユーザが指定するGUIの配置条件を考慮し、GUIの位置、大きさ等を調整し、表示画面908に送出する。

【0106】以下に上記の実例の例を示す。

【0107】図26に実際の表示画面でのオブジェクトのGUIの形態の概念を示す。オブジェクトにより作成されたGUIは常時画面に表示されるわけではなく、「表示」と「非表示」という二つの状態をとる。システムが立ち上がった時点の状態は、非表示状態とする。ある時点でのネットワーク上の全ての装置のオブジェクトのGUIはこの二つの状態に関して同一状態をとるものとする。この「表示」と「非表示」という二つの状態は受信装置900本体の制御パネルや附属のリモート・コントロール・ユニット等のボタンによってトグルとなるようにしても良い。

【0108】表示状態となったGUIの表現は二通りで、「アイコン」の形態か「制御パネル」の形態である。システムが立ち上がった時点の表示形態は、アイコン形態とする。また、非表示状態から表示状態へ移行したときアイコン形態とするようにしても良い。

【0109】各オブジェクトのGUIは、それぞれ独自の表示形態（アイコン／制御パネル）を持つことが可能である。GUIの表示状態が非表示状態となった場合は、これまで表示されていた各ネットワーク装置のGUIの表示形態（アイコン／制御パネル）、表示位置、表示の大きさ等がエンジン906に全て記憶され、再び表示状態となった場合にそれらの情報を元に、元通りに表示される。これは受信装置900の電源が落とされても

有効である。再び、受信装置900が立ち上がると、以前に接続していて、かつ今も立ち上がっている装置については以前の状態で表示する。

【0110】図27にアイコンの表示例について示す。アイコンは表示画面908において、左上隅から左下隅方向へ順番に並ぶ。最下端まで到達したら、一列右にずれて、また上から下に並ぶ。以後はこれを繰り返す。アイコン間のフォーカスの移動にはカーソル5000を用いる。カーソル5000は各アイコン間を離散的に移動するものとする。アイコン化のオン/オフは、カーソル5000を移動し、そのアイコンを選択してから決定ボタンを押す等により決定する。

【0111】このオブジェクトのGUIの配置条件としては、所定の分類（例えば、VTR、ディスプレイ、DVD等の機器毎の分類や、記録機器、再生機器等の機能的分類等）により分類し、順番に表示するようにしても良い。

【0112】図37はこのように機器毎に分類して表示した例である。表示画面908の左上隅から左下隅方向にかけて該分類毎に順番に並んでいる。この表示するときの分類は、予め設定されている他、ユーザがどのような分類にするのかを設定変更できるようになっていれば更に便利である。このように分類毎に分かれて順番に表示されているので、ユーザはGUIから所望のアイコンの選択がし易くなる。

【0113】また、任意のネットワーク機器については予めGUIを表示することを停止できるよう予め設定しておけるようにすることも可能である。これにより、ユーザは表示画面908により制御することのないであろう機器についてはアイコン表示されないで、必要とするアイコンのみ表示させることが出来、アイコンの選択等がし易くなり使い勝手が向上する。この分類毎に順番に表示することについては、以降全ての実施の形態に適用することが出来る。

【0114】受信装置900自身についてのGUIについては、ネットワーク機器の一つとして他のネットワーク機器と同じくGUIを表示するようにしても良いし、表示しないようにしても良い。自身についてGUIを表示する場合、同じ操作環境を実現することができる。また、自身についてGUIを表示しない場合、リモコン等

でいちいちGUIを選択することなく自身を直接制御することができる。

【0115】図28に制御パネルについて一例を示す。制御パネルは装置の制御を行なうためのGUIを提供する。受信装置900の表示画面908には、ストップボタン5010、プレイボタン5011等が表示され、これらを例えばカーソルにより選択し決定操作を行うことによって、この決定された制御命令を制御パネルの送出元のネットワーク機器に送出することができる。また、スライド式の音声ボリューム5012も表示され、この

送出元のネットワーク機器の音声の大きさを調整する機能を提供する。これらのGUIはビットマップ等の画像を用いたり、何らかの図形描画言語を用いたものを利用する。この図形描画言語とは、文字多重放送で用いられているジオメトリック符号や、Javaのグラフィックライブラリ等も含む。図示しないが複数の制御パネル間の移動に関してもアイコンの規則と同様にカーソルによる選択/決定の方式が取られ、制御パネル内のボタン等の移動/決定も同様である。

【0116】この制御パネルの表示時、この制御パネルのネットワーク機器が例えばプレイ動作を行っていたときにはプレイボタン5011の枠が赤色となり、現在プレイ状態であることが判別可能に表示されている。ここでカーソルをストップボタン5010に移動、決定し該機器を停止させたときにはストップボタン5010の枠が赤色となり、停止動作となったことが分かる。また、音声の大きさを調整する機能についても同様に、現在の音声の大きさに応じた位置にスライドが存在するように表示されている。これらは、ネットワーク機器が自身の動作をネットワークへ送出することにより実現される。

【0117】また、受信装置900が2台繋がったネットワークでは、2台の受信装置900から同じ機器に対して同時に異なる操作がされてしまうと問題がある。このようなことが発生するのを防ぐためには予め2つの受信装置900に予め優先順位を付けておいたり、あるネットワーク機器に対して同時に複数の受信装置900に制御パネルを表示できないようにすれば良い。

【0118】また、制御パネルの中に該ネットワーク機器のGUIを入れて表示することにより、どのネットワーク機器のGUIなのか知ることが出来、誤った機器に対して制御することを防ぐことが出来る。このようにすることについては、以降全ての実施の形態にも適用することが出来る。

【0119】図29～図32には制御パネルとアイコンを同時に表示した例を示している。

【0120】図29の例は、互いの制御パネル5021、5022、5023がお互いに重ならないように、かつアイコンの場所と重ならないよう、その配置、大きさを設定して画面左下から水平方向右に並べられる。また最下段が埋まったら、一段上にずれて並べられる。この図以降、制御パネルを枠としてしか記載していないものについては省略して記載しているだけであり、実際には図28に示すようなネットワーク機器に応じた制御パネルが表示される。

【0121】制御パネルが表示された後、この制御パネルに対応するアイコンを表示画面908に表示し続けるのか、消すようにするのかについては、どちらであっても良い。表示し続ける場合には、アイコン周囲の背景（例えばアイコン周囲を縁取る線）と制御パネルの背景とを同色とすることにより、どのアイコンの制御パネル

10

20

30

40

50

なのかが判別し易くなる。また、制御パネルを表示させた後にもアイコンを表示し続ける場合には、表示されているアイコンの形を変形させたり、背景色を変える等して、既に制御パネルが表示されていることがアイコンで判断出来るようにすることも出来る。

【0122】逆に、消えるようにした場合、表示画面908がアイコンや制御パネルで埋め尽くされにくくなるので、ユーザはアイコン／制御パネルの後ろに表示されている映像を良く見ることが出来、表示画面908を有効に利用することが出来る。このように制御パネルが表示されたときに対応するアイコンを表示画面から消す実施の形態は、以降全ての実施の形態にも適用することが出来る。

【0123】図30の例は、複数の制御パネル5021、5022、5023が重なることを許す例である。制御パネル5021、5022、5023は互いに少しずつつらして画面左下に重ねて表示する。隠れている制御パネル5022、5023は可視部分をクリックすることによってその制御パネルを最前面表示し、目的とする制御パネルへフォーカスを移すことができる。

【0124】図31の例は、常に必ず一つだけしか制御パネル5024の表示を許さず、他はアイコン化する例である。アイコンの1つを選択すると、画面左下に選択したアイコンに対応する制御パネル5024が表示され、これにフォーカスが移り、現在表示されている制御パネルはアイコン化され、入れ替わるようになっていく。この制御パネルの状態は、ユーザによる新たな制御が無い時間が所定時間（例えば5分間）続いたとき、自動的にアイコンに変わるようになっていく。これにより誤ってリモコン等を押してしまったときの制御パネルの誤制御を防ぐことが出来る。

【0125】図32の例は、最高n個までの制御パネルの表示を許す例を示している。n個未満の制御パネルが表示されている状態で、あるアイコンを選択しアイコン状態を解除すると、このアイコンに対応する制御パネルが図29或いは図30のように新たな制御パネルとなって表示される。また、n個未満の制御パネルが表示されている状態で、ユーザがあるアイコンを選択しアイコン状態を解除すると、図31の例のように既表示の制御パネルから制御パネルの一つを選んで、当該制御パネルに対応するアイコン化表示を行い、新たに制御パネルとして前面に表示されたものにフォーカスが移る。

【0126】既表示の制御パネルの中からアイコン化する制御パネルを選択する方法としては、図示しない以下のようなものがあげられる。

【0127】1. 最も最近制御パネルを表示したn-1個の制御パネルと、新たに選択した一つの制御パネルを表示する。

【0128】2. 最も最近利用した、つまり制御パネル内部のコマンドを使用した、n-1個の制御パネルと、

新たに選択した一つの制御パネルを表示する。

【0129】3. 現在制御パネルが表示されているものの中で、最も最近利用していないn-1個の制御パネルと、新たに選択した一つの制御パネルを表示する。

【0130】各制御パネルの位置、大きさはユーザのインタラクションにより自由に制御される。また、制御パネルを構成するボタン等のコンポーネントの形状等はビットマップの取り替え等により自由に変更可能である。

【0131】図示しないが、あるネットワーク機器が電源が落されるなどの理由で、利用不可となった場合、対応するGUI（アイコンもしくは制御パネル）は表示画面から削除され、利用不可となったことを示すメッセージを表示画面908に所定の期間表示する。

【0132】図33を参照してこの発明の他の実施の形態を説明する。

【0133】上記のようにネットワーク装置、例えばDVDプレーヤー952とVTR951がIEEE1394のようなネットワークを通じてエンジン906に接続される。この場合、上記の例であると、それぞれのネットワーク機器にアイコンや制御パネルの表示用のオブジェクトが記憶されているとした。しかしこれに限らず各種の方法が可能である。

【0134】例えばDVD952は、その内部にそのIDとともに機能一覧6010を持ち、この中には例えば4種類の機能 Play（再生）、Stop（停止）、Forward Skip（前方ヘスキップ）、Backward Skip（後方ヘスキップ）の4種類の機能が各々00、01、04、05と符号化されて保持されている。同様にVCR（ビデオ・カセット・レコーダ、或いはVTR：ビデオ・テープ・レコーダ）951は、そのIDと共に機能一覧6011を持ち、この中には機能 Play（再生）、Stop（停止）、Feed Forward（巻き取り）、Rewind（巻き戻し）が各々00、01、02、03と符号化されて格納されている。

【0135】エンジン906にはこれらの機能に対応するGUIオブジェクト6013が格納されており、エンジン906は機能一覧を参照して知ったDVDとVCR各々の持つ機能に対応するGUIオブジェクトを表示画面908に表示する。

【0136】表示画面908上で、例えばVCR951の Play に対応するGUIオブジェクト（ボタン）6014がユーザによって押されると、エンジン906はVCR951に対してコマンド00（Play）をIEEE1394のようなネットワークを通じて送信する。

【0137】このような構成をとることによって、DVD952、VCR951等各々のネットワーク機器がGUIオブジェクトを保持する場合に比べ、ネットワーク機器上のメモリの節約になると共に、意味的に同じ機能（例えばVCRの Play とDVDの Play 等）が画面上で異なったGUIとなるのを防ぐことが可能になってい

る。つまりユーザにとって意味する記号（GUI）に統一性が取れることになる。

【0138】さらに、図に示した機能を示す番号および参照の方法をすべてのエンジン、ネットワーク機器について共通にすることにより、相互接続性を保証することが可能になっている。

【0139】従来においては、家庭における家電機器の制御操作は本体に装備されている制御ボタン、制御パネル等を用いて行なわれることは稀で、各々を個別のそれ専用の附属のリモート・コントロール・ユニットを用いて制御するのが一般的である。しかし、各々のリモート・コントロール・ユニットはその本体の制御を行なうのに特化されており、他の装置の操作は行なうことができないことが多い。例え一つのリモート・コントロール・ユニットが他の装置の制御を行なえたとしても、実現できる機能は一般的な機能に限定され、その装置固有の制御を行なうことはできない場合が多い。また、各装置がTVなどの画面表示装置と映像ケーブル等を用いて接続されることによって、表示装置にその装置のGUIを表示できるようになっている。しかし、各々のGUIは他の装置の表示する同様なGUIとの関連性を考慮していないことが多く、時には両者が表示装置の画面上でオーバーラップして見にくくなってしまいう状態が発生する。

【0140】しかしこの発明によると上記のような問題が解消される。

【0141】この発明は、上述した実施の形態に限定されるものではない。

【0142】イベント（Event）の形式や取り扱い方式が異なる場合も、当該イベントに回答できるようにエンジンが工夫されている。以下、これを実現するためのデータ変換器について説明する。

【0143】まず、データ変換が必要なことを理解するため、実行方式1（第1のエンジン方式）と実行方式2（第2のエンジン方式）のイベントの形式及び取り扱い方式について説明する。

【0144】図34（A）は第1のエンジンによる実行方式1（第1のエンジン方式という）におけるイベントの取り扱いを示しており、イベントは、プロセデュアル・コード（Procedural Code）の呼び出しと直接関連づけられている。プロセデュアル・コードは、図22のバーチャルマシン（VM）909上で実行されるプログラムであり、例えばBASICやJavaといった計算機言語で記述されたものを用いることができる。

【0145】実行方式1ではイベントの発生とプロセデュアル・コードの呼び出しとは等価である。図では例としてプログラム（Program）という名称のプロセデュアル・コード7011中でEvという名称のプロセデュアル・コード7012が呼び出される場合を示しているが、このような呼び出しの操作がイベントに相当することになる。呼び出しの際にはデータを引数として渡すこ

とが許されており、図ではp1、p2を渡している。この場合、イベントにはイベントデータp1、p2が付随していることになる。イベントはアプリケーションの実行においてアプリケーション内部で種々の動作を起動させたり、アプリケーションを構成する各オブジェクトの状態を変化させたりするために用いられる。例えば、ある時刻になったことを示すためにイベントが発生し、これに付随するイベントデータは例えば時刻データであり、呼び出されたプロセデュアル・コードの内部で時刻に応じた処理を行うために渡される。

【0146】図34（B）は、第2エンジンによる実行方式2（第2のエンジン方式）におけるイベントの扱いを示している。イベントの役割や用いられ方は、上記した第実行方式1の場合と同様であって、実行方式1のそれと同様にデータを付随させることもできるが、イベントはオブジェクトからオブジェクトへ送信されるメッセージとして扱われることや、イベントが型を持っていることが実行方式1と異なっている。イベントの型は、そのイベントが何故生じたかを示すために用いられ、例えば“ある時刻に到達した”、“あるボタンが押された”、などを識別するために用いられる。

【0147】この実行方式2におけるイベントは、実行方式1のイベントのようにプロセデュアル・コード7012を直接呼び出すことはできず、必ずリンクオブジェクト（Link Object）7021へ送られる。リンクオブジェクト7021は第2のエンジン方式のアプリケーションを構成するオブジェクトの1種である。

【0148】リンクコンディション（Link Condition）7022は受け取ったイベントに対する条件を記述したものでありイベントソース（Event Source）7024、イベントタイプ（Event Type）7025、イベントデータ（Event Data）7026から構成されている。

【0149】リンクオブジェクト7021がイベントを受け取ると、イベントの発生源（送信元オブジェクト）とイベントソース7024、イベントの型とイベントタイプ7025、付属データとイベントデータ7026がそれぞれ比較される。これにより、例えばどのアイコン（ソース）のどのボタン（タイプ）がどのように操作される（データ）かを示すことになる。

【0150】比較の結果、一致しないものがあればそのイベントに関する動作は終了する。全てが一致した場合には、リンクエフェクト（Link Effect）7023が起動される。リンクエフェクト7023は要素動作（エレメンタリーアクション；Elementary Action）7027の並びである。具体的な要素動作の例には、他のオブジェクトへイベントを送るSENDイベント（Send Event）や、プロセデュアル・コードを呼び出すコール（Call）などがある。

【0151】上記のように実行方式1と実行方式2ではイベントの形式や取り扱い方式が異なるため、実行方式

1のアプリケーションを直接的に実行方式2のアプリケーションに変換することはできなかった。

【0152】しかし本発明では図35に示すようにプロセデュアル・コードの呼び出しである実行方式1のイベントから実行方式2のイベント8011とリンクオブジェクト8012の組を生成するように工夫している。リンクオブジェクト8012のリンクコンディション8013にはイベント8011によって常に成立する条件を記述する。そしてリンクエフェクト8014にはプロセデュアル・コード8016を呼び出す要素動作8015 (Call) を記述することにより、実行方式1のイベントによってプロセデュアル・コードが実行されるのと等価な動作を実行方式2で実現している。

【0153】このような処理を実現することにより、本体が実行方式2のタイプであっても、実行方式1のイベントに対応することができる。

【0154】次に、図36を参照して、実行方式2のアプリケーションを実行方式1のアプリケーションに変換する場合のイベントの取り扱いを説明する。つまり本体が実行方式1のタイプに対して、実行方式2のイベントが

入力された場合、このイベントの扱い方を説明する。【0155】実行方式2では型と付属データを伴ったイベント8041がリンクオブジェクト8042へ入力されるが、このイベント8041に対応するものとして、実行方式1の、呼び出し用のイベント8044を生成する。このイベント8044は、引数として、どこで発生したかを示すソース (Source) と、その型 (Type) と、付属データ (Data) を持つことになる。このイベント8044は、プロセデュアル・コード8043の呼び出し用として機能する。つまり、イベント8041に対応させて引数を持つイベント8044を発生させる。この図の例えば、Source=1, Type=0, Data=2という引数に変換されている。

【0156】次に、プロセデュアル・コード8043は、実行方式2のリンクコンディション8045に記述された条件を判定する引数比較部8046と、第2のエンジンのリンクエフェクト8047に記述された動作を実現する動作記述部8048として構成する。

【0157】引数比較部8046の実現方法としては、図に示す例の場合、図36の符号8049に示すような条件分岐構文による記述がある。例えば、

```
IF (Source != 1) return
(もしソースが1でないならば戻りなさい)
IF (Type != 0) return
(もしタイプが0でないならば戻りなさい)
IF (Data != 2) return
(もしデータが2でないならば戻りなさい)
```

である。

【0158】このように記述することで条件が一致すれば (Source=1, Type=0, Data=2)、引数比較部

8046を通過し、動作記述部8048に進むことになる。これにより、Source=1, Type=0, Data=2の条件に対応したアクション1、2、3が実行に移される。この実行は、各要素動作 (アクション; 実行方式2) と等価なコード (実行方式1) を予め作成しておき、これを図36の符号8050のように配列しておくことにより実現できる。

【0159】以上のように本発明ではリンクオブジェクトと等価な動作を行うプロセデュアル・コードを実行方式1用に生成することで等価な動作を実現している。

【0160】以上説明したこの発明によると以下の番号にそれぞれ述べるような各特徴を備えている。

【0161】(1) 実行するアプリケーションを、1つあるいは複数の信号伝送システムに対応する、映像、音声、データとして送信する装置において、前記アプリケーションを構成する要素を、インタラクティブ機能を有する異なるテレビジョンシステムで共通に用いる基本的な共通データと、おのおの異なる信号伝送システムが個別に用いる個別データとに分解して送信する。

【0162】(2) また、上記の信号伝送システムから伝送された信号を受信する装置であって、前記共通データのみを用いてアプリケーションを実行するか、または、前記信号送信装置の出力を受信し、前記共通データおよび、自身の方式に対応した前記個別データとを合成してアプリケーションを実行するか、または、前記信号送信装置の出力を受信し、前記共通データおよび、自身の方式に対応した前記個別データ、更に、ひとつ以上の自身の方式と異なる方式に対応した前記個別データとを合成してアプリケーションを実行する。

【0163】(3) 前記共通データおよび個別データは、アプリケーションの構成を記述したオブジェクトと、所定の符号化方式に従ってエンコードされたコンテンツと、制御動作を記述したスクリプトのデータの種類の持つ。

【0164】(4) 前記個別データは、共通データの性質を基本的に有し、個別のテレビジョンシステムに独自で共通データに不足する詳細な情報である補足データと、共通データの性質を基本的に有しないスクリプト、コンテンツデータである独自データとからなる。

【0165】(5) 上記共通データ中の3つのデータの種類の属する各データ単位は、自身を識別するための第1の識別子を持つ。

【0166】(6) また、個別データ中の3つのデータの種類の属する各データ単位は、自身の基本的な性質を持つ共通データを知るための第2の識別子を持つこととなる。

【0167】(7) 更に独自データであって、そのデータ単位が持つ第2の識別子が、空データのような第4のデータの種類のデータ単位の第1の識別子と関連付けられ、更に異なる方式の伝送受信に対しても拡張性を有す

10

20

30

40

50

るように図られている。つまり共通データがない方式のものも空の共通データを設けるようにし、前記拡張性を持たせている。

【0168】(8)更に共通データは、自身を識別する第1識別子のみを持つ。つまり空の共通データであれば、自身を識別第1の識別子をもてばオブジェクトやスクリプトのための識別子を持たなくてもよい。

【0169】(9)また共通データと個別データを合成するにあたり、個別データの独自データはそのスクリプト、コンテンツを有効成分とし、第4のデータの種類の共通データ(空の共通データ)は、その第1の識別子を有効成分として合成することになる。

【0170】(10)前記個別データは、共通データの性質を基本的に有し、個別のテレビジョンシステムに独自で共通データに不足する詳細な情報である補足データと、共通データの性質を基本的に有しないスクリプト、コンテンツデータである独自データとからなり、前記個別データ中の前記補足データのデータ単位は、自身の基本的な性質を持つ共通データを知るための第2の識別子を持ち、同独自データのデータ単位は、自身を識別するための第1の識別子を持つ事を特徴とする。

【0171】(11)第2の識別子は、所属するデータ単位の基本的な性質を持つ共通データのデータ単位の第1の識別子と同じ値を持ち、共通データと個別データの性質の関係を把握できるようにしている。

【0172】(12)共通データは、複数の異なるテレビジョンシステムにより生成されたアプリケーション群から、いずれのテレビジョンシステムに対しても独立となる規則により生成される。

【0173】(13)共通データの性質を基本的に有し、個別のテレビジョンシステムに独自で共通データに不足する詳細な情報である補足データの生成および共通データとの合成は、オブジェクト指向技術のクラス継承技術に基づきなされる。

【0174】(14)前記アプリケーションの基本となる構成を記述したオブジェクトとは、実行時に画面上に表示される可視性オブジェクト、画面上に表示されない非可視性オブジェクト、アプリケーション全体を制御する大局的な性質を持つオブジェクト、オブジェクト間の制御を行う内部的な性質をもつオブジェクトにより構成される。

【0175】(15)共通データに属するオブジェクトは、オブジェクトを識別する識別子、オブジェクトのタイプを示す識別子、データコンテンツを指定する識別子、スクリプトを指定する識別子をもつ、また、共通データに属するコンテンツは、コンテンツを識別する識別子、コンテンツのタイプを示す識別子をもつ。そしてまた、共通データに属するスクリプトは、スクリプトを識別する識別子、スクリプトのタイプを示す識別子をもつ。

【0176】(16)第1のテレビジョンシステムに適合するアプリケーションのための第1の個別データと第1の共通データを受信し、前記第1の個別データを変換することにより、第2のテレビジョンシステムのアプリケーションに必要な第2の個別データを生成し、この第2の個別データと前記第1の共通データとを用いて第2のテレビジョンシステムに適合するアプリケーションを実行する手段を有する。

【0177】(17)第1のテレビジョンシステムに作られたアプリケーションより生成された個別データ1と共通データと、第2のテレビジョンシステムのアプリケーションに必要な個別データ2を前記個別データ1より変換により生成し、前記共通データと個別データ1、個別データ2を送信することができる。

【0178】(18)個別データの変換は、2つ以上の複数のテレビジョンシステム間により行われる。

【0179】(19)個別データの変換は、2つ以上の複数のテレビジョンシステム用の個別データから、一つのテレビジョンシステム用の個別データを生成することを特徴とする。また個別データの変換は、1つのテレビジョンシステム用の個別データから、複数のテレビジョンシステム用の個別データを生成することを特徴とする。更にまた個別データの変換は、2つ以上の複数のテレビジョンシステム用の個別データから、一つのテレビジョンシステム用の個別データを生成することを特徴とする。

【0180】(20)共通データ、および個別データは、複数の異なるテレビジョンシステムにより生成されたアプリケーション群から、いずれのテレビジョンシステムに対しても独立となる規則により生成される。

【0181】(21)実行するアプリケーションを、1つあるいは複数の方式に対応する、映像・音声・データとして送信する場合、前記アプリケーションを構成する要素を、インタラクティブ機能を有する異なる方式で共通に用いる基本的な共通データと、前記異なる方式で各々個別に用いる個別データとに分解して送信する信号送信装置からの信号と、自身を操作するために用いるデータであって、前記インタラクティブ機能を有する異なる方式に変換可能なフォーマットに従ったデータを送信する機能を有する1つ以上の装置からの信号を受信する受信装置であって、前記共通データ、前記変換可能なフォーマットに従ったデータ、前記個別データ、のいずれか1つあるいは、前記1つ以上のデータの合成データを用いてアプリケーションを実行することを特徴とする信号受信装置。

【0182】(22)前記変換可能なフォーマットに従ったデータを用いたアプリケーション実行の結果として、該データの送信元である送信機の制御を行うデータを送出することができる。

【0183】(23)前記共通データは、前記変換可能なフォーマットに従ったデータのフォーマットに従って

記述されている。

【0184】(24)(21)に記載された装置であって、動作条件と動作記述とから構成されリンクオブジェクトへイベントが入力され、このイベントが動作条件を満たしていた場合にのみ動作記述で示された動作を行う第2のエンジン方式のアプリケーションを、イベントの発生とプロセデュアル・コードの実行とが等価である第1のエンジン方式のアプリケーションへと変換する変換器において、第2のエンジン方式のイベントを、そのイベントに付随するデータを引数とする第1のエンジン方式のプロセデュアル・コードの呼び出しに変換するイベント変換手段と、第2のエンジン方式のリンクオブジェクト中の動作条件を、前記呼び出しの対象となるプロセデュアル・コード中の引数比較部に変換する動作条件変換手段と、前記リンクオブジェクト中の動作記述を前記プロセデュアル・コード中の動作記述部へと変換する動作記述変換手段とを備える。

【0185】(25)引数比較部は、第2のエンジン方式のリンクオブジェクト中の動作条件を判定の条件とし、この条件と引数として渡されたデータの内容とを比較し条件が成立するかどうかを判定する。

【0186】(26)プロセデュアル・コード中の動作記述部は、第2のエンジン方式のリンクオブジェクト中の動作記述と等価な動作が第1のエンジン上で行われるように記述されている。

【0187】(27)動作記述変換手段は、第2のエンジン方式のリンクオブジェクト中の動作記述の構成要素となりうるすべてあるいは一部の記述の各々について、それらと等価なプロセデュアル・コードを予め生成しておき、それらを参照しながら動作記述部を生成する。

【0188】(28)(21)に記載の装置であって、イベントの発生とプロセデュアル・コードの実行とが等価である第1のエンジン方式のアプリケーションを、イベントは動作条件と動作記述とから構成されるリンクオブジェクトへ入力されイベントが動作条件を満たしていた場合のみ動作記述の動作を行う第2のエンジン方式のアプリケーションへ変換する変換器において、第1のエンジン方式のイベントを、第2のエンジン方式のイベントとリンクオブジェクトの組に変換することを特徴とする。

【0189】(29)上記(28)の第2のエンジン方式のイベントとリンクオブジェクトの組の生成にあたっては、そのリンクオブジェクトは、常に成立する動作条件と、プロセデュアル・コードを呼び出す動作記述とから構成される。

【0190】(30)(21)の装置において各々のネットワーク機器は自身が備えている機能の一覧を保持し、エンジンはこの一覧データを参照し、各々の機器に対して行うことのできる操作の種類を知る手段を持つ。

【0191】(31)(30)において各々のネットワ

ーク機器の保持する機能の一覧のエンジンに参照は、エンジンの種類に関わらず一定のインタフェースによって行われることを特徴とする。

【0192】(32)(30)においてエンジンは、ネットワーク機器が備えている機能のそれぞれに対応するグラフィカル・ユーザ・インタフェイス(GUI)オブジェクトを保持し、ネットワーク機器が備えている機能についてこれを表示画面上でユーザに提示し、提示されたGUIオブジェクトに対するユーザの操作をネットワーク機器に対して行うことを特徴とする。

【0193】(33)(21)記載の装置は、自身を操作するために用いるデータのグラフィカル・ユーザ・インタフェイス間の相対的配置関係を調整する手段を有することを特徴とする。

【0194】(34)上記(33)の装置を実現する手段として、データを新たに受信したことを検知して、検知したことを表すグラフィックスを送出する手段と、合成データのグラフィカル・ユーザ・インタフェイスの位置、大きさを検出する手段と、その情報を元に、グラフィカル・ユーザ・インタフェイスの配置を調整する手段とを有することを特徴とする。

【0195】

【発明の効果】以上説明したようにこの発明によれば、それぞれ固有に開発された双方向システムに対し相互に互換性を確保することが可能な装置を得ることができ

る。
【0196】また各装置にその固有の制御情報を予め所有させて、それらの情報を集中的に管理し、統一的な制御操作を行なうことができる。また、デジタルテレビジョン放送などの電波放送においてデータの転送が考えられることも踏まえ、それらのGUIオブジェクトと各装置の制御のためのGUIとの統一的な画面表示の調整を行なうこともできる。

【図面の簡単な説明】

【図1】この発明の一実施の形態を示す図。

【図2】この発明の他の実施の形態を示す図。

【図3】図1のブロックの具体的構成例を示す図。

【図4】この発明のデータのクラスの形態例を示す図。

【図5】この発明の装置の動作機能を説明するために示した説明図。

【図6】この発明の更に他の実施の形態を示す図。

【図7】この発明のまた更に他の実施の形態を示す図。

【図8】この発明の装置の表示例を説明するために示した説明図。

【図9】この発明によるデータの伝送形態例を説明するために示した図。

【図10】この発明によるデータの他の伝送形態例を説明するために示した図。

【図11】この発明によるデータの更に他の伝送形態例を説明するために示した図。

【図12】この発明で用いられるデータ単位の識別信号の例を示す図。

【図13】従来のボタンオブジェクトの例を示す説明図。

【図14】従来のボタンオブジェクトの他の例を示す図。

【図15】この発明のボタンオブジェクトの例を示す図。

【図16】従来のビットマップの例を示す説明図。

【図17】従来のビットマップの他の例を示す説明図。

【図18】この発明のビットマップの例を示す図。

【図19】この発明によるデータの更に他の伝送形態例を説明するために示した図。

【図20】この発明によるデータの更に他の伝送形態例を説明するために示した図。

【図21】この発明によるデータの更にまた他の伝送形態例を説明するために示した図。

【図22】この発明のさらに他の実施の形態を示す図。

【図23】図22の装置の表示画面の例を示す図。

【図24】この発明に係るネットワーク装置の構成例を示す図。

【図25】図24のエンジンにおける画面調整装置の構成例を示す図。

【図26】画面の表示形態のマップを示す図。

【図27】表示画面の表示例を示す図。

*【図28】同じく表示画面の表示例を示す図。

【図29】同じく表示画面の表示例を示す図。

【図30】同じく表示画面の表示例を示す図。

【図31】同じく表示画面の表示例を示す図。

【図32】同じく表示画面の表示例を示す図。

【図33】ネットワーク装置の制御パネル表示の例を示す図。

【図34】実行方式の異なるイベントのタイプおよび取り扱いの説明図。

10 【図35】実行方式1のイベントを実行方式2で扱うための変換処理の説明図。

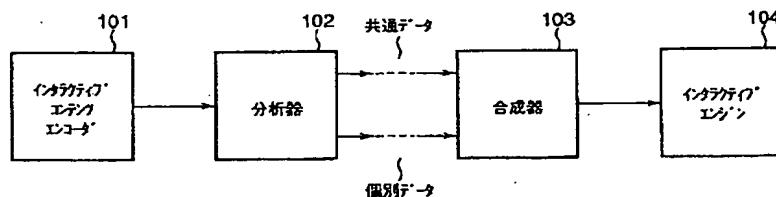
【図36】実行方式2のイベントを実行方式1で扱うための変換処理の説明図。

【図37】表示画面の表示例を示す図。

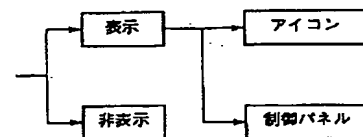
【符号の説明】

101、202…インタラクティブコンテンツエンコーダ、102、203…分析器、103、208…合成器、104、209…インタラクティブエンジン、201…AVエンコーダ、204…フォーマットエンコーダ、205…多重器、206…分離器、207…フォーマットデコーダ、210…AVデコーダ、904…合成器、905…インターフェイス、906…エンジン、907…変換器、908…表示画面、909…バーチャルマシン、920…入出力端子、951、952、953…ネットワーク装置、960…ネットワーク。

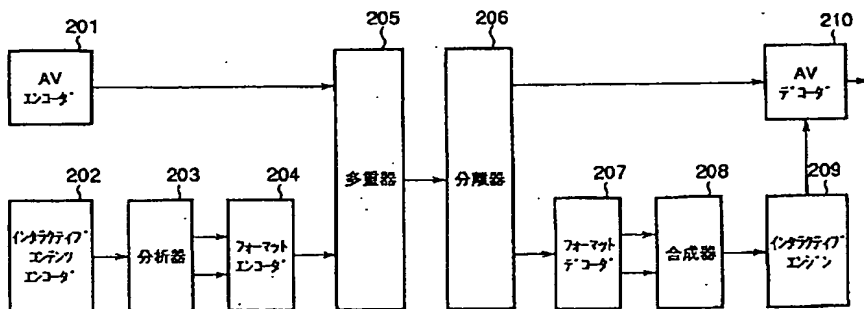
【図1】



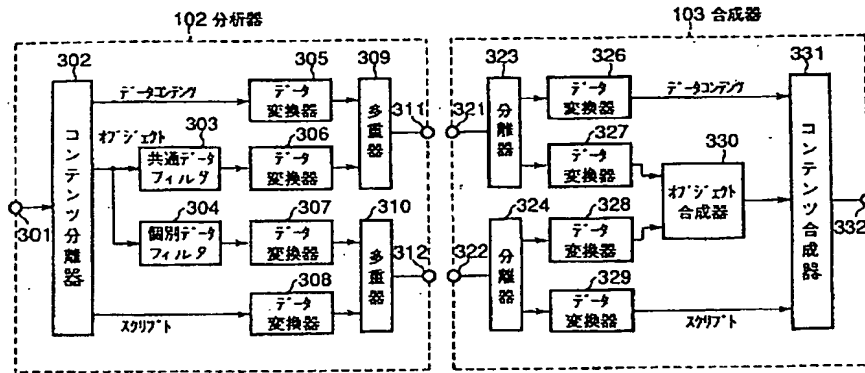
【図26】



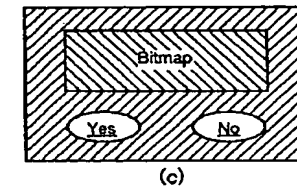
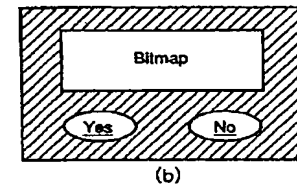
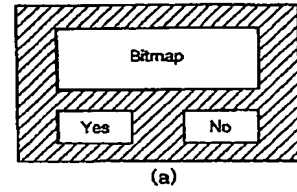
【図2】



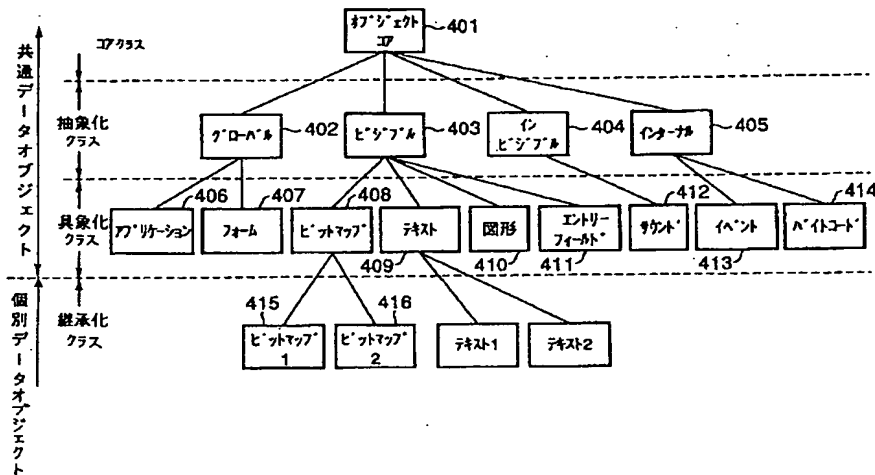
【図3】



【図8】



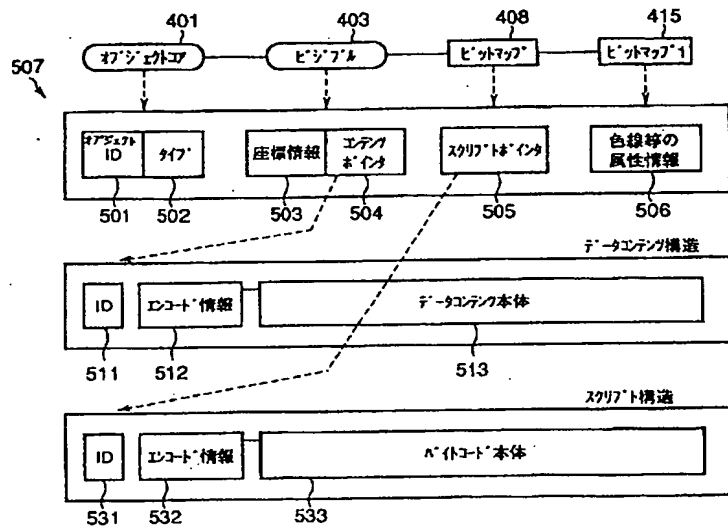
【図4】



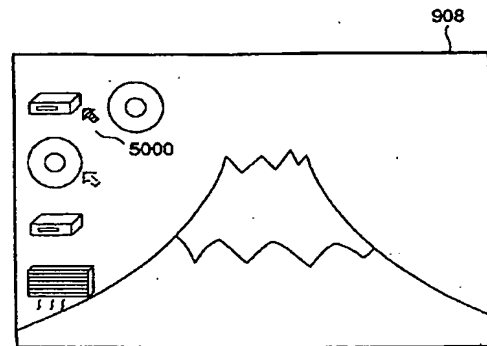
【図9】

構成要素		情報		
		engine 1 (共通)	engine 2 (差分)	engine 3 (差分)
objects (オブジェクト)	Scene	位置、形状、枠線、背景色	—	—
	Event	振る舞い	—	—
	Bitmap (1)	位置、素材番号	—	半透明
	Button (2)	位置、素材番号	横円、下線	—
	Button (3)	位置、素材番号	横円、下線	—
contents (コンテンツ)	Bitmap (1)	素材本体	—	—
	Text (2)	素材本体	—	—
	Text (3)	素材本体	—	—
script (スクリプト)	Script A (2)	—	振る舞い	—
	Script A (3)	—	振る舞い	—

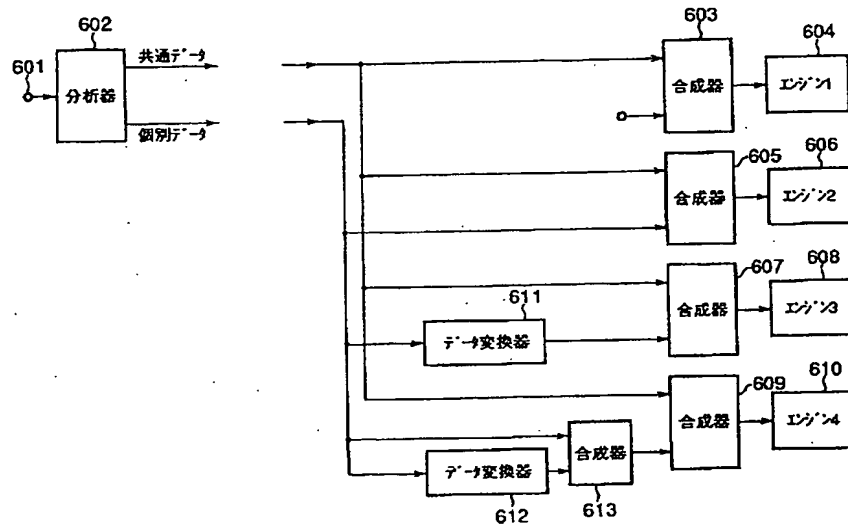
【図5】



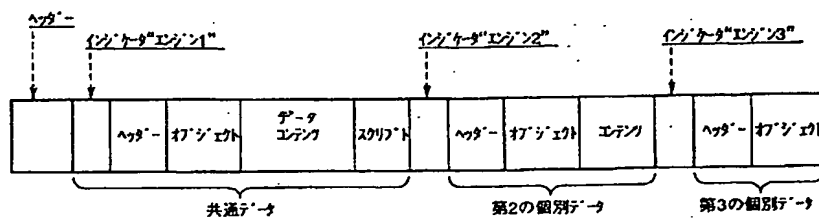
【図27】



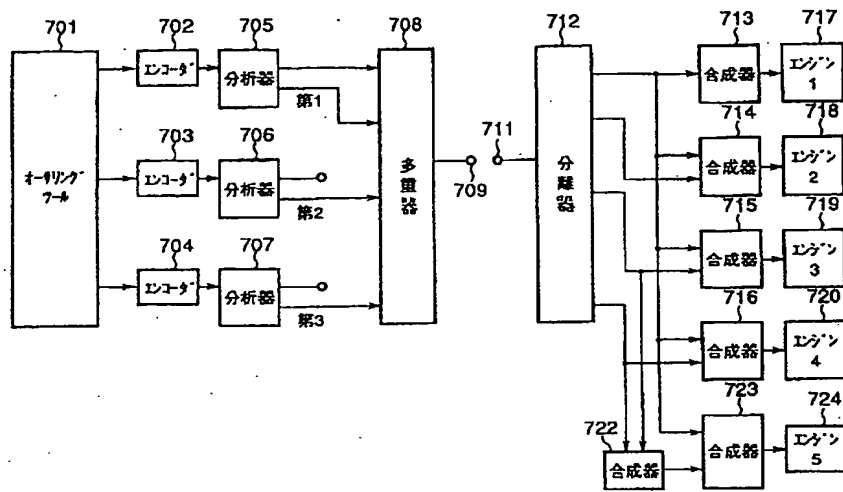
【図6】



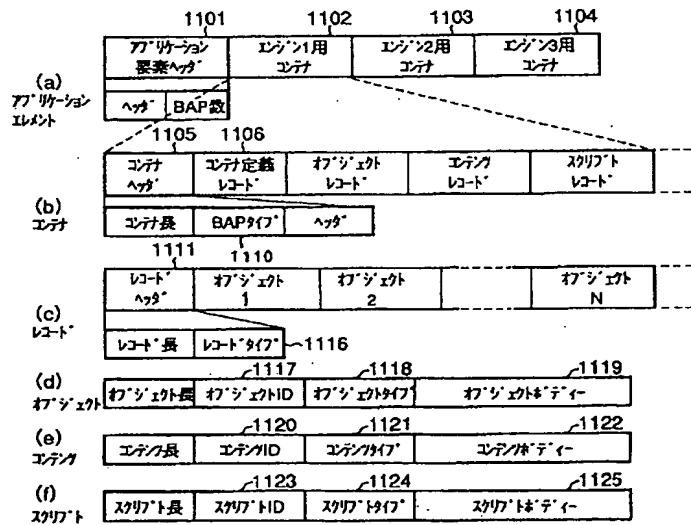
【図10】



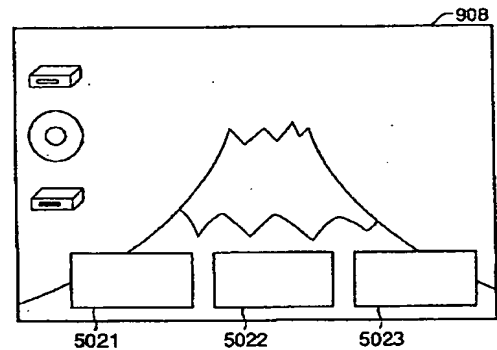
【図7】



【図11】



【図29】



【図12】

		オブジェクト (objects)	コンテンツ (contents)	スクリプト (script)
コンテナ (container)	エンジン1用 (共通)	自分の識別 (ID)	自分の識別 (ID)	自分の識別 (ID)
	エンジン2用 (差分)	自分の所属 (ID)	自分の識別 (ID)	自分の識別 (ID)

(*) i>0の整数

IDの意味付け

【図13】

```

ICAPButtonObject := (
  hintFlag
  objectType      ; オブジェクトの識別子
  position         ; オブジェクトの左上の位置
  extent          ; オブジェクトの右下の位置
  horz_align
  vert_align
  resource         ; オブジェクトへのポインタ
  trigger_script
  hints : HINT_ROUND_KEYCAP ; ボタンを丸く表示することを示すフラグ
)

```

エンジン2用のButtonオブジェクトの構成

【図15】

(共通要素)

```

CommonButtonObject (
  objectID      ; オブジェクトの識別子
  UpperLeftPosition ; オブジェクトの左上の位置
  LowerRightPosition ; オブジェクトの右下の位置
  ContentReference ; オブジェクトへのポインタ
)

```

(差分要素2)

```

Engine02ButtonObject (
  horz-align
  vert-align
  trigger-script
  hints : HINT_KEYCAP_ROUND
)

```

(差分要素3)

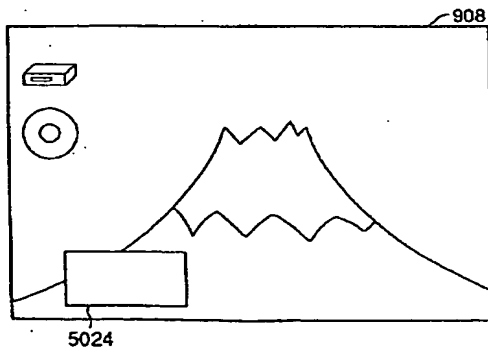
```

Engine03ButtonObject (
  no-caching
  initially-active
  content-hook
  content-size
  shared
  engine-resp
  highlight-ref-colour
  interactive-palette-ref
  ref-colour1
  ref-colour2
)

```

buttonオブジェクトの構成

【図31】



【図14】

```

ButtonClass := (
  object-identifier ; オブジェクトの識別子
  no-caching
  initially-active
  content-hook
  content-data      ; オブジェクトへのポインタ
  content-size
  shared
  x-length          ; オブジェクトの水平方向の大きさ
  y-length          ; オブジェクトの垂直方向の大きさ
  x-position        ; オブジェクトの水平方向の位置
  y-position        ; オブジェクトの垂直方向の位置
  engine-resp
  highlight-ref-colour
  interactive-palette-ref
  ref-colour1
  ref-colour2
)

```

エンジン3用のButtonオブジェクト構成

【図16】

```

ICAPBitmapObject := (
  hintFlag
  objectType      ; オブジェクトの識別子
  position         ; オブジェクトの左上の位置
  extent          ; オブジェクトの右下の位置
  horz_align
  vert_align
  resource         ; ビットマップデータへのポインタ
  hints
)

```

エンジン2用のBitmapオブジェクトの構成

【図17】

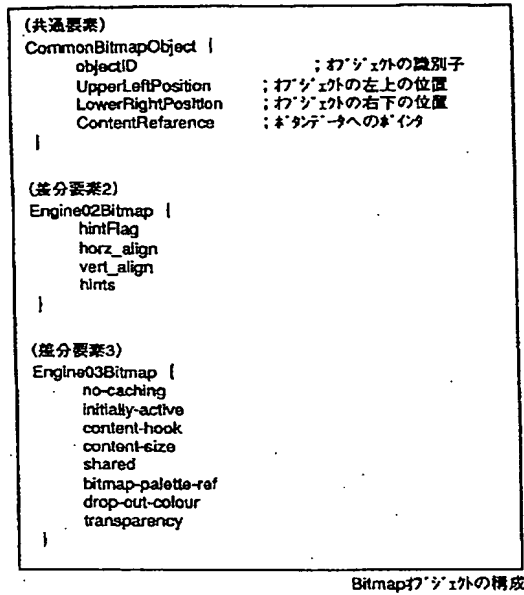
```

BitmapClass := (
  object-identifier ; オブジェクトの識別子
  no-caching
  initially-active
  content-hook
  content-data      ; ビットマップデータへのポインタ
  content-size
  shared
  x-length          ; オブジェクトの水平方向の大きさ
  y-length          ; オブジェクトの垂直方向の大きさ
  x-position        ; オブジェクトの水平方向の位置
  y-position        ; オブジェクトの垂直方向の位置
  bitmap-palette-ref
  drop-out-colour
  transparency      ; ビットマップを半透明で表示することを示すフラグ
)

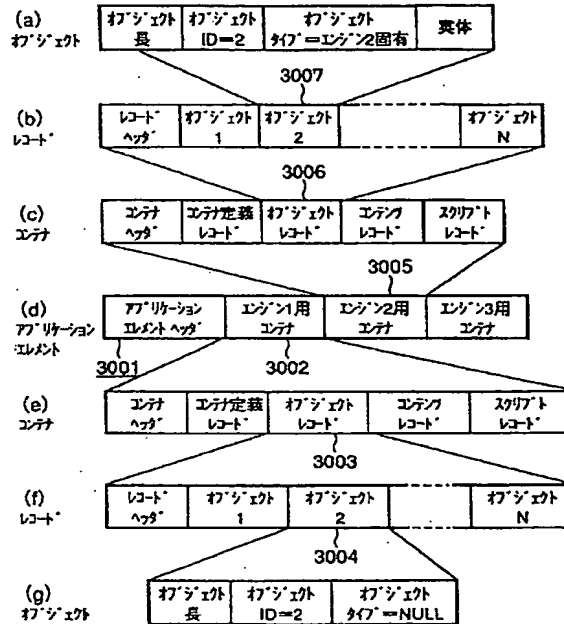
```

エンジン3用のBitmapオブジェクト構成

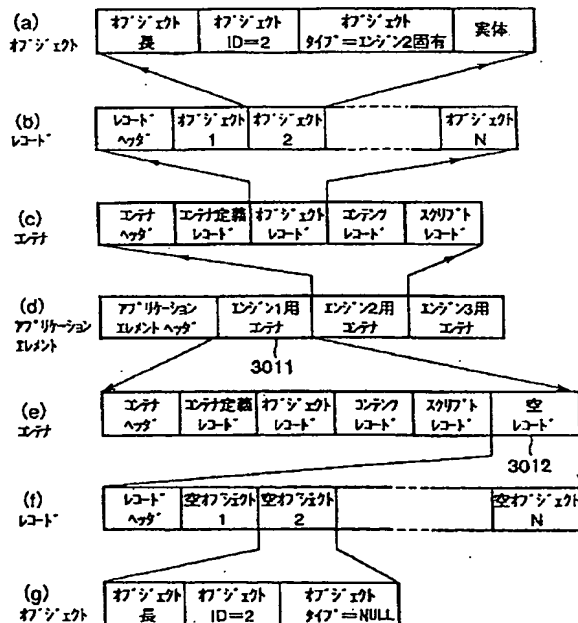
【図18】



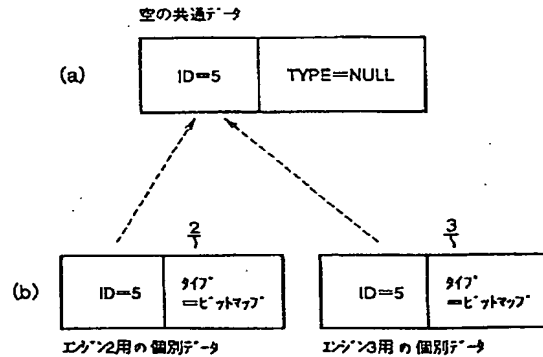
【図19】



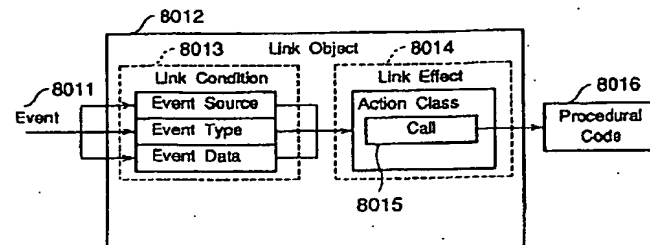
【図20】



【図21】

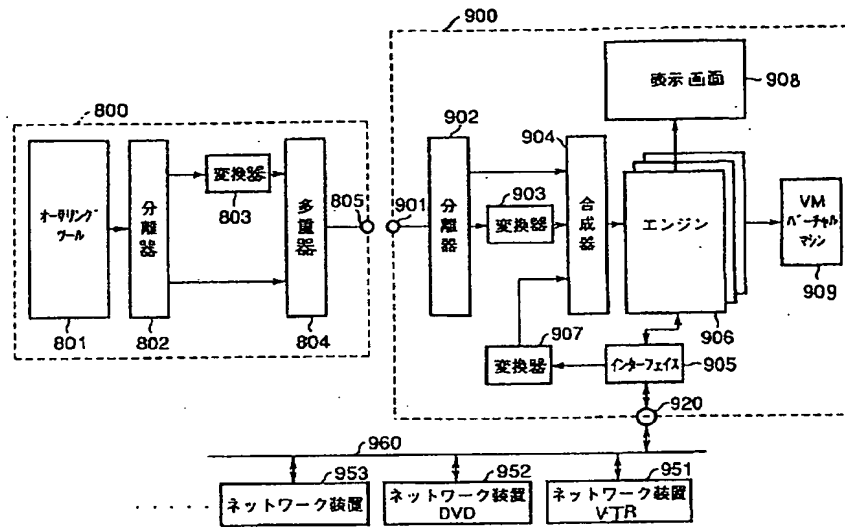


【図35】

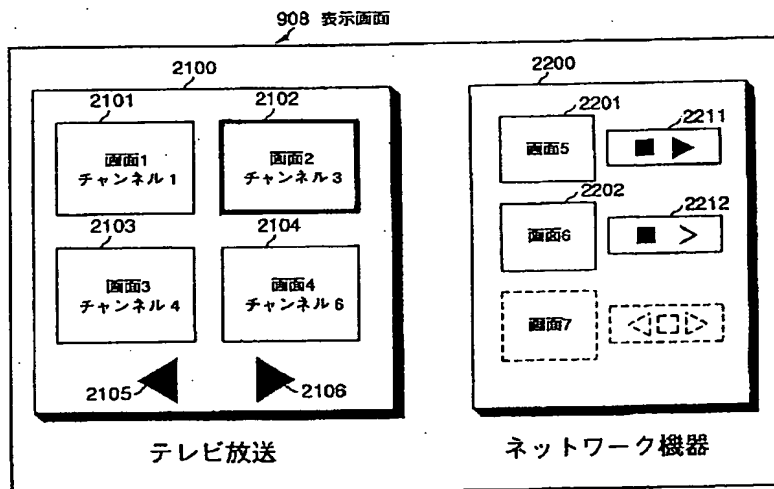


エンジン方式のEventをエンジン2方式に変換

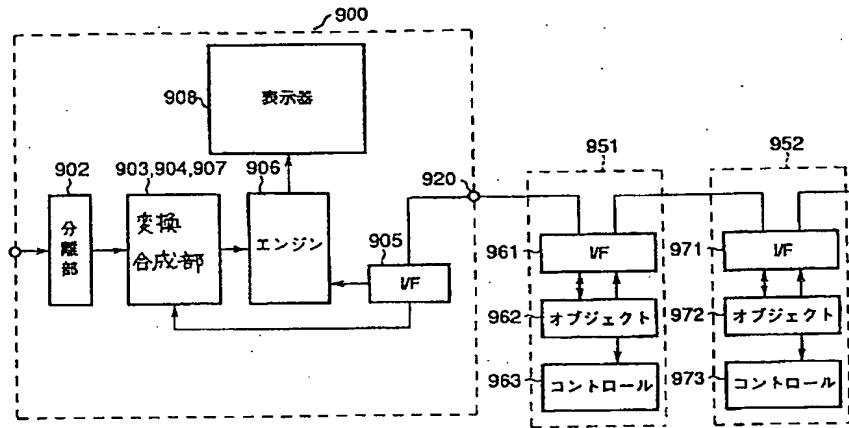
【図22】



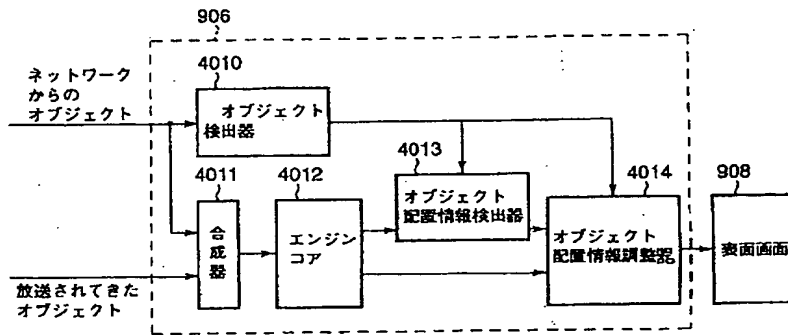
【図23】



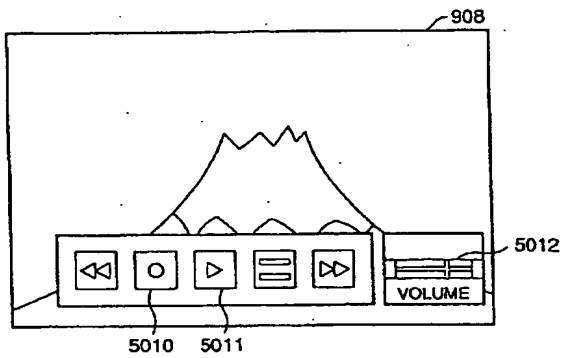
【図24】



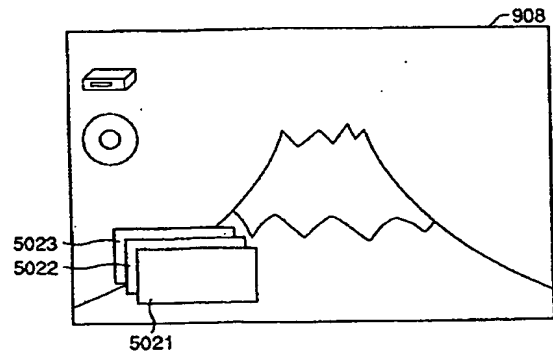
【図25】



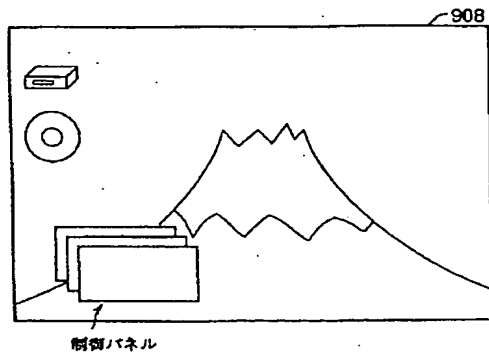
【図28】



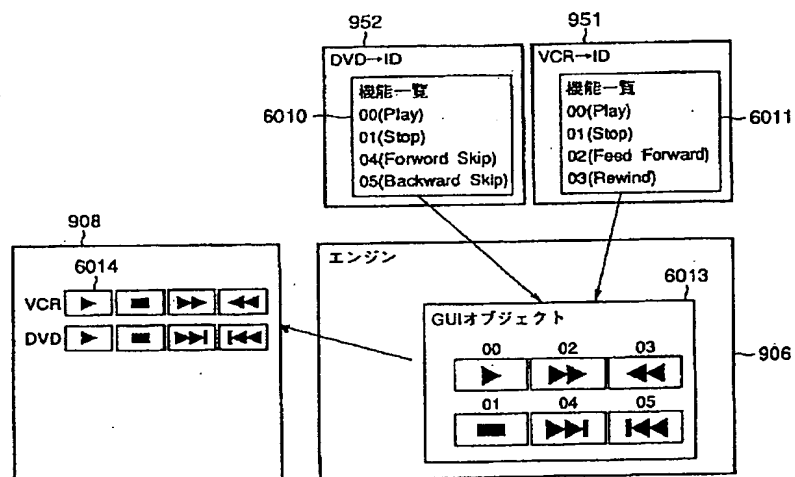
【図30】



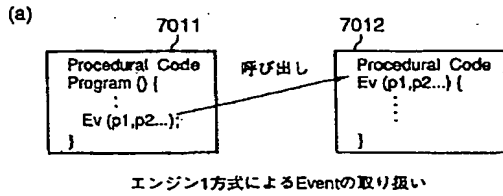
【図32】



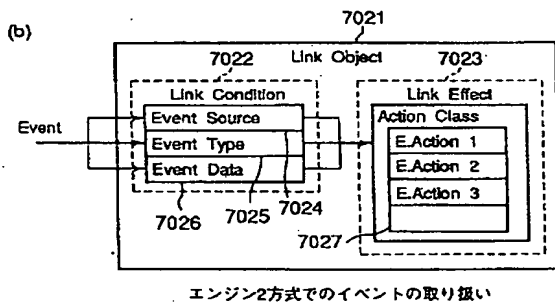
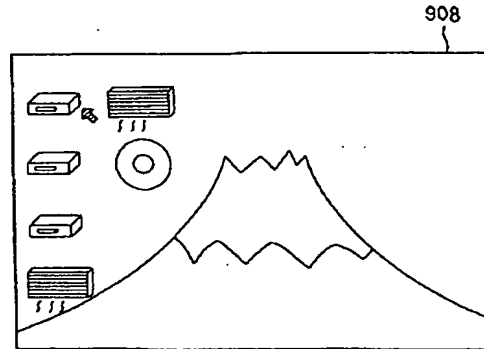
【図33】



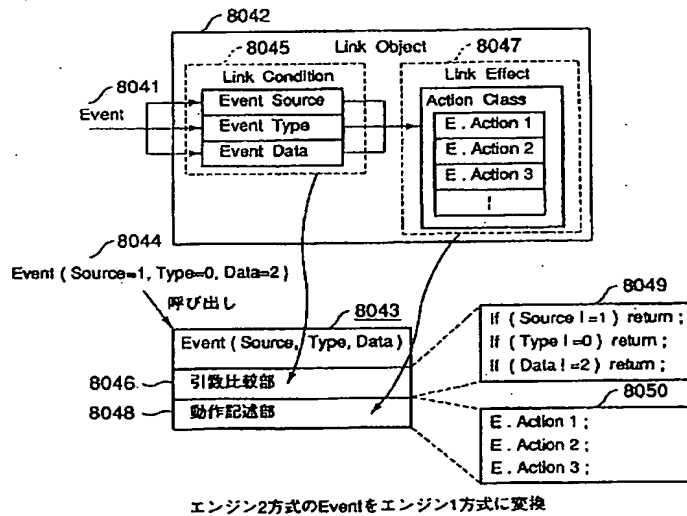
【図34】



【図37】



【図36】



フロントページの続き

(72)発明者 村松 孝倫
神奈川県横浜市磯子区新杉田町8番地 株
式会社東芝マルチメディア技術研究所内

(72)発明者 下田 乾二
神奈川県横浜市磯子区新杉田町8番地 株
式会社東芝マルチメディア技術研究所内